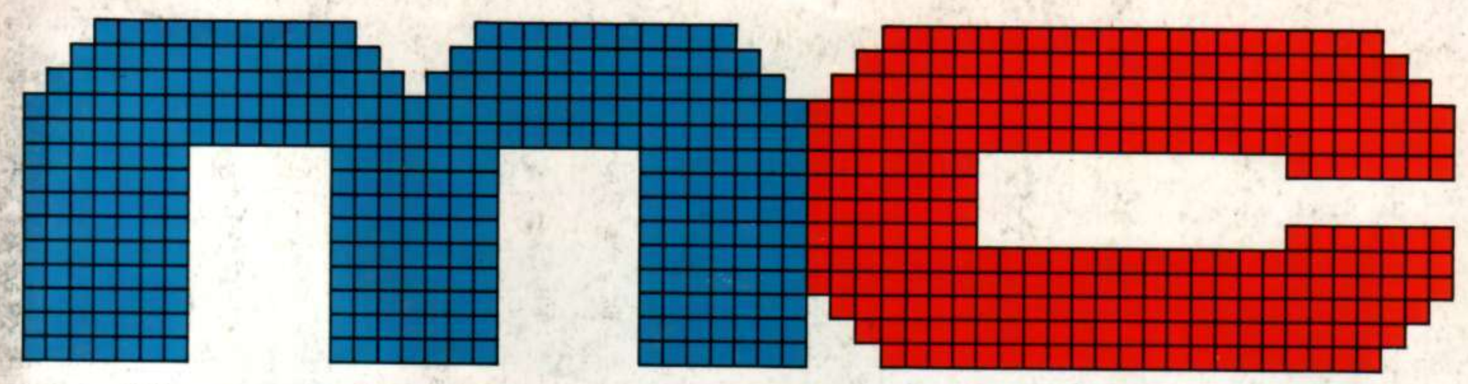


Februar 1988
7 DM · 60 öS · 7 sFr

Die Mikrocomputer-Zeitschrift

2/88

PC-Betriebssystem:

MS-OS/2 hebt ab

DOS- und PC-Technik:
Timer richtig nutzen

Batch-Dateien jetzt wirklich vollautomatisch

Kompaktkurs:
Modula-2 für Einsteiger

Atari ST:
Profi-Tastatur selbst gebaut

Omikron-Compiler

Alles auf einen Blick:
Super-Tabelle für PCs mit ROM-BIOS-, Ein-/Ausgabe-, RAM- und Video-Referenzen

Test:
80386 von Osborne

Lisp-Interpreter

EPSON. Der Unterschied.



Automatisches Einzelblatt-Magazin als Option.

Selbst unser kleinster 24-Nadel-Drucker bietet Schönschrift auf höchstem Niveau. Der neue EPSON LQ-500.

Die 24-Nadel-Drucker der EPSON LQ-Serie haben Zeichen in der Schriftqualität gesetzt. LQ = Letter Quality wurde zum Begriff für echte Schönschrift, die höchsten Ansprüchen genügt. LQ steht aber auch für hohe Druckleistung bei äußerst niedrigen Anschaffungs- und Betriebs-Kosten. Diese Unterschiede in Qualität und Wirtschaftlichkeit unterstreicht der neue EPSON LQ-500. Mit einem überraschend günstigen Preis-/Leistungs-Verhältnis.

Dieser Text ist in der klassischen Schönschrift Roman geschrieben worden.

Hier ist ein Muster der modernen Schönschrift Sans Serif.

Für einen äußerst attraktiven Preis bietet er hochauflösende Grafik, gut lesbare Schnell- und zwei perfekte LQ-Schönschriften. In vielfältigen Varianten, darunter Großschrift, Outline- und Shadow-Schrift. Das hohe Drucktempo und die reichhaltige Ausstattung erfüllen alle professionellen Anforderungen. Der EPSON LQ-500 wird sowohl Aufsteiger im Home-Bereich als auch Freiberufler und mittelständische Anwender begeistern.



Technologie, die Zeichen setzt.

EPSON Deutschland GmbH · Zülpicher Straße 6 · 4000 Düsseldorf 11 · Telefon 0211/5603-0
Vertriebsbüro Hamburg: Telefon 040/4413 31-34 · Vertriebsbüro München: Telefon 089/9172 05-07

M.L.&S.

EDITORIAL

OS/2 hebt ab. In aller Stille sind von den neuen IBM-/2-Modellen vergangenes Jahr große Mengen über den Ladentisch gegangen. Noch nicht mit OS/2 ausgestattet, aber immerhin ...
Es ist also alles für den Start von OS/2 bereit, der in diesem Frühjahr offiziell stattfinden wird. Wir bei mc haben für Sie schon vorab Einiges ausprobieren können. Lesen Sie, was unser Spezialist Dieter Strauß herausgefunden hat.



Liebe Leser!

Modula-2 ist die Fortsetzung von Pascal mit höheren Mitteln. Wir antworten auf die vielen Leserbriefe, die sich mit GOTO und dem strukturierten Programmieren beschäftigen, jetzt mit einer Serie über die Programmiersprache Modula. In das Konzept dieser Sprache ist alles eingearbeitet, was das „wohlformulierte", moderne Programmieren ausmacht. Die Serie beginnt mit der Formulierung der Prinzipien der strukturierten Programmierung, damit jeder weiß, worüber gesprochen wird. Nochmals sei gesagt, daß es nicht darum geht, irgendeine Programmiersprache als gut oder schlecht zu qualifizieren. Allerdings ergeben sich größte Vorteile, wenn man die Techniken nutzt, die Modula in Fortsetzung von Pascal bietet – gerade bei größeren Programmen. Das alles bleibt nicht etwa Theorie: Auf dem Markt erscheinen immer mehr preiswerte Compiler für den PC, den Atari ST und andere Computermodelle. Das beweist ein Blick in den Anzeigenteil von mc.

Die Mischung in dieser Ausgabe ist breit. Zum Beispiel bringen wir für Leser mit Interesse für die „absoluten" Grundlagen den Beitrag „Prolog und Logik". Die Beweismechanismen von Prolog werden hier vor dem Hintergrund der Prädikatenlogik verständlich. Ein Thema, das einen wichtigen Aspekt der künstlichen Intelligenz abdeckt und gleichzeitig mit den Grundlagen des Denkens überhaupt verknüpft ist. In dieser Ausgabe gibt es auch einen Beitrag zum mc-modular-AT, der allen an der PC-Technik Interessierten zeigt, wie ein Festplatten-/Floppy-Controller im AT installiert ist. Ein Beitrag schildert, wie man einen Z80 mit dem Amiga von Commodore verheiratet. In unserer Testabteilung finden Sie einen „386er" von Osborne getestet, Lisp-Interpreter untersucht, den neuen Sanyo-AT mit 10-MByte-Floppy (!) zur Besichtigung und einen schnellen Basic-Compiler für den Atari ST in Betrieb genommen.

Eine Riesenmenge Ataris aller Klassen sind verkauft. Geschickt ist hier das Marketingziel – ein Computer, der von Profis und von Amateuren akzeptiert werden kann – umgesetzt worden. Was aber alle engagierten Benutzer beklagen, ist die Konstruktion der Tastatur: Die sei leider sehr „heimcomputerig". mc hilft dem ab. Mit mc können Sie Ihren Atari ST mit einer neuen Tastatur ausrüsten. Mit etwas Know-How kommen Sie so zu einer preiswerten Profitastatur mit Siemens-Tasten.

Service für die Leser ist eines unserer Hauptziele. Diesmal finden Sie eine große Tabelle, die erschöpfend Auskunft über die Internas von Original und Kompatiblen gib. Viele Abweichungen „nicht ganz" Kompatibler sind mit enthalten. Eine Tabelle, die man sich gut aufheben sollte – so kompakt finden Sie diese Referenzen nirgendwo. Viel Nutzen aus der umfangreichen mc-PC-XT-AT-Referenzliste (/2-Busbelegungen eingeschlossen) wünscht Ihnen

Ihr
Ulrich Rohde

INHALT

INFO

GRUNDLAGEN

SOFTWARE

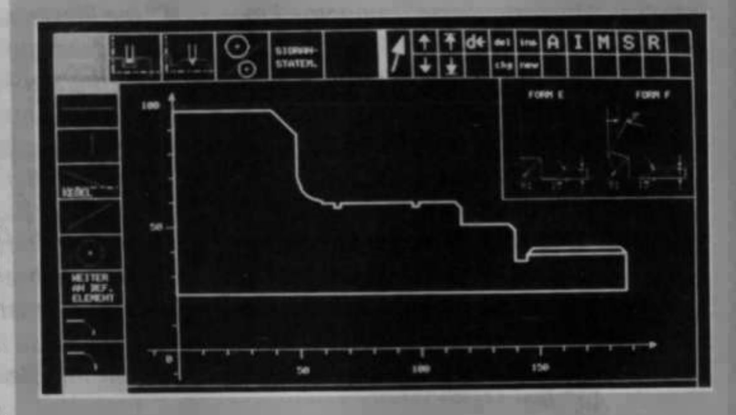
MS-OS/2 hebt ab



Titelfoto: Klaus Hager

Das neue Betriebssystem für Rechner der AT-Klasse kommt in Kürze auf den Markt. Obwohl es MS-DOS noch sehr lange geben wird und Microsoft sich verpflichtet hat, MS-DOS weiterhin zu unterstützen, lohnt es sich, einen Blick auf MS-OS/2 zu werfen und sich über dessen Eigenschaften zu informieren. Dem Anwendungs-Programmierer öffnet sich eine breite Palette an Möglichkeiten, Programme noch effektiver zu gestalten. Wir machen mit neuen Begriffen wie z. B. dem Session-Manager, den Familien- und Kernel-Applikationen und den Threads vertraut.

Seite 38



Digitale Bilderzeugung und -verarbeitung

Leistungsfähige Software setzt flexible und schnelle Hardware voraus. Die sich immer stärker verbreitenden grafischen Bedieneroberflächen erfordern moderne Grafik-Prozessoren. Wir stellen den ACRTC 63484 und den GDP 82786 vor. Dabei wird auf den komplexen internen Aufbau und den Befehlssatz eingegangen.

Seite 44

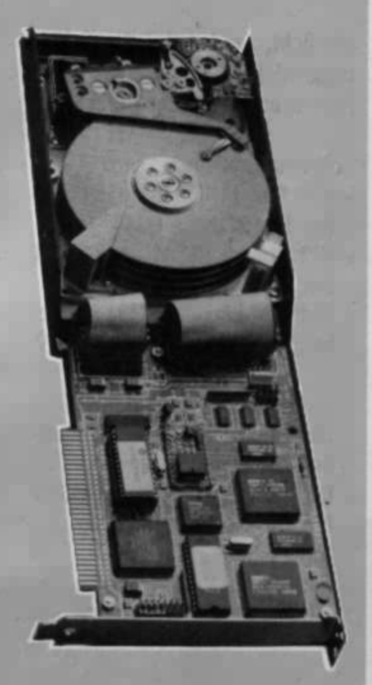


Die mc-ST-Tastatur

Wer mit dem Tastatur-Computer Atari 1040 ST oder 520 ST arbeitet, hat sich sicher schon über die Tastatur aufgeregt. Diese ist nicht wie bei jedem modernen Rechner frei beweglich, da alle Peripheriegeräte direkt mit der Tastatur verbunden sind. Mit unserer Umbauanleitung, die genau beschreibt, wie es geht und worauf man achten muß, wird dieser Umstand ad acta gelegt.

Seite 112

Festplatte sicherer gemacht



Vor Datenverlust ist niemand gefeit. Auch die beste und teuerste Festplatte ist kein Garant dafür, daß über kurz oder lang ein Fehler auftritt, der wichtige und vielleicht sogar unwiederbringbare Daten zerstört. Es ist scheinbar nur eine Frage der Zeit, wann die latente Gefahr ausbricht und für einen erhöhten Adrenalin-Ausstoß sorgt. Mit dem Programm Disk Technician, das es jetzt für MS-DOS-PC gibt, ist die Gefahr zwar nicht zu bannen, aber doch zumindest deutlich zu reduzieren. **Seite 124**



Z80-Schnittstelle für den Amiga

Die Rechner der Amiga-Familie zeichnen sich durch eine offene System-Architektur aus. Alle internen Signale werden an einer leicht zugänglichen Schnittstelle für Erweiterungen angeboten. Mit einem minimalen Aufwand an Hardware können die Amiga-Rechner mit dem NDR-Klein-Computer-Bus verbunden werden. Ein Artikel für alle, die sich für den Amiga interessieren. **Seite 108**



80386-Power von Osborne

Viel Computer für sein Geld bekommt man mit dem O6T-386 von Osborne. Besonders bei den Massenspeichern Diskette und Festplatte zeigt sich der Computer flexibel. Sei es nun ein 3½-Zoll-Disketten-Laufwerk mit 720 KByte oder eines mit 1,44 MByte – bei Osborne gibt es alles. Wir beschreiben die Speicher-Architektur dieses Rechners ausführlich. **Seite 117**

INHALT

HARDWARE

TEST

RUBRIKEN



Die Mikrocomputer-Zeitschrift

Franzis-Verlag GmbH
Karlstraße 37–41, 8000 München 2
Postfach 37 01 20, 8000 München 37
Telefon (0 89) 51 17-1
Telex 5 22 301
Telefax (0 89) 51 17-3 79
Tedas (0 89) 59 84 23 und 59 64 22

Chefredakteur:
Dipl.-Math. Ulrich Rohde

Stellvertretender Chefredakteur:
Dipl.-Ing. (FH) Ulrich Kruppe

Redaktion:
Redaktionsassistentin:
Rita Schleser (0 89) 51 17-3 54
Ressortredakteure: Horst Brand, Dipl.-Ing. (FH) Wolfgang Hascher (fl), Dipl.-Ing. (FH) Rudolf Hofer (fl), Reiner Schönrock, Dipl.-Ing. (FH) Dieter Strauß

Ständiger Mitarbeiter:
Dipl.-Ing. (FH) Herwig Feichtinger (zu erreichen unter der Anschrift der Redaktion)

Korrespondent: USA:
Stan Baker, 504 Nino Avenue, Los Gatos, CA 95030

Art Direktion:
Dipl.-Designer (FH)
Volker Hilbel

Layout, Grafik, Herstellung:
Josef Wurzinger

Franzis-Labor:
Neucom electronic GmbH

Software Service:
SHAMROCK Software Vertrieb
(0 89) 59 54 68

Verantwortlich für den Inhalt:
Dipl.-Math. Ulrich Rohde

Verantwortlich für Anzeigen:
Dietger Kötter

Gesamtherstellung:
Franzis-Druck GmbH
Karlstraße 35
8000 München 2
Telefon (0 89) 51 17-1

ISSN: 0720-4442



Vertriebskennzeichen:
B 7745 E

Sonderdrucke:
Jakob Wintersberger

Lizenzen:
Georg Geschke, Königstr. 8,
3000 Hannover 1,
Telefon (05 11) 34 53 62

BRIEFE

Hochfrequenz

Es erschüttert mich, daß Sie und Ihre Redakteure nicht wußten, daß ein Hochfrequenzgerät einer Zulassung durch die Bundespost bedarf. Nur: Eine Einzelgenehmigung ist sicher auch nicht der Weisheit letzter Schluß, ist sie doch an den Aufstellungsort des Gerätes gebunden. Der Buchstabe des Gesetzes sagt in diesem Fall, daß der Computer mit Einzelgenehmigung zwar im Wohnzimmer, aber nicht im Büro oder in der Computerwerkstatt um die Ekke (!) betrieben werden darf. Die fehlende Genehmigung kann auch bei Kompatiblen einiger Billiganbieter zum Problem für den Anwender werden.
Sie sollten bei Tests und Marktübersichten deshalb auch immer anmerken, ob für das vorgestellte Gerät eine (generelle) Genehmigung der Deutschen Bundespost vorliegt. Wer einen direkt importierten Computer betreiben will, muß auch auf alle Fälle die Genehmigung dazu beantragen. Die Ordnungsstrafe kann (finanziell) recht empfindlich ausfallen, und das widerrechtlich betriebene Gerät kann eingezogen werden.

Jürgen Lange
6230 Frankfurt

Anmerkung der Red.: Daß komplett angebotene Geräte einer solchen Genehmigung bedürfen, ist sicher gut und richtig. Daß man aber einen selbstgebauten UKW-Empfänger (der ja durchaus HF-Schwingungen erzeugt) legal an die Dachantenne anschließen kann, mit dem selbstgebauten Rechner im HF-dichten Metallgehäuse jedoch gegen geltendes Recht verstößt, erscheint uns wenig logisch...

Basic-EMUF

Im mc Heft 12/1986 haben Sie den Bauvorschlag für den Basic-EMUF veröffentlicht. Ich habe mir nach dem abgedruckten Layout eine Platine angefertigt und den CPU-Teil und die RTC bestückt. Leider lief die Schaltung nicht, da sich auf dem Layout einige Fehler befinden.

Fehlende Verbindungen:
- IC2 Pin 11 (T1in) nach IC4 Pin 11 (CONS OUT)
- IC2 Pin 12 (R1out) nach IC4 Pin 10 (Cons IN)
- Auf der Lötseite IC3, Leiterbahn von Pin 10 muß an Pin 9

Falsche Verbindungen:
- Ubat an R 10 mit +5V verbunden (Bestückungsseite)
- D3 am IC10 Pin 15 mit ST5 Pin 28 (GND) verbunden

Nach diesen Änderungen lief mein Basic-EMUF. Über den restlichen Peripherie-Teil kann ich jedoch keine Aussage machen. Die Fehlersuche hat mich doch ganz schön geärgert; zeitweise hatte ich schon fast den Spaß am Basic-EMUF verloren. Jedoch hat nicht jeder Hobbybastler das Wissen, um eine solch fehlerhafte Platine zum Laufen zu bringen. Wenn dann mindestens eine Bastelleiche herumliegt, in die man viel Geld investiert hat, kann einem bald der Spaß am Selberbauen vergehen.Ich habe bis jetzt erst einen Bauvorschlag nachgebaut, der keinen Fehler enthielt.
Ansonsten kann ich nur sagen, daß ich Ihre Zeitschrift für die Beste auf diesem Gebiet halte und Sie hoffentlich noch viele interessante Artikel veröffentlichen (hoffentlich auch fehlerfrei).

Peter Foschum
7900 Ulm

Monte Carlo im Computer

Die von C. Jäkel vorgeschlagene Methode zur Erzeugung von normalverteilten Zufallszahlen mit Hilfe der Poisson-Verteilung ist nicht optimal. Bei Anwendung des im Bild als Basic-Programm gezeigten Verfahrens ergibt sich eine saubere, elegantere und schnellere Lösung: In einem ersten Schritt wird eine Zufallszahl gebildet und in den aufgrund des Mittelwertes und der Standardabweichung festgelegten Bereich umgerechnet. Hier sind die Zahlen zunächst noch gleichverteilt. Mit Hilfe einer zweiten Zufallszahl wird sie nun bewertet. Nur wenn sie unterhalb der Verteilungsfunktion liegt, wird sie verwendet. Da sich die Normalverteilung über alle reelen Zahlen erstreckt, muß der Bereich eingegrenzt werden: hier auf ±4∗Sigma, was für alle praktisch vorkommenden Fälle ausreichen dürfte.
Im Bereich S1...S2 werden N normalverteilte (Pseudo-)Zufallszahlen mit Mittelwert MY und der Standardabweichung SIGMA ausgedruckt. Bei 1000 Zahlen mit dem Mittelwert 10 und der Standardabweichung 2 liegen im Bereich 2...18 >99,99% aller zu erwartenden Zahlen. Das Verfahren ist exakt und kann entsprechend modifiziert auch für andere Verteilungen verwendet werden.

Dr. G. Born
8011 Vaterstetten

```
100 PRINT "ZUFALLSZAHLEN NORMALVERTEILT"
110 PRINT "============================"
120 INPUT "WIEVIEL ZAHLEN......";N
130 INPUT "MITTELWERT BEI......";MY
140 INPUT "STANDARDABWEICHUNG..";SIGMA
150 S1=MY-4*SIGMA
160 S2=MY+4*SIGMA
170 MO=2*SIGMA*SIGMA
180 I=0
190 REM REPEAT
200 : GOSUB 1000        :REM  1.ZUFALLSZ.
210 : Z1=S1+ZUF*(S2-S1):REM  AUS S1...S2
220 : PHI=EXP(-(Z1-MY)*(Z1-MY)/MO)
230 : GOSUB 1000        :REM  2.ZUFALLSZ.
240 : Z2=ZUF            :REM  AUS  0...1
250 : IF Z2<PHI THEN PRINT Z1 : I=I+1
260 IF I<N THEN 190 : REM UNTIL
270 END
1000 REM ZUFALLSZAHLENGENERATOR
1010 : ZUF=RND(1)
1020 RETURN
```

Normalverteilte Pseudo-Zufallszahlen gewinnen



Wolf K.
Dipl.-Ing.
Karin A.
Sekretärin
Helmut M.
EDV-Berater
Stimmt!
Ihr Drucker-Füchse!
Jetzt kommt der Super-Sprinter unter den Qualitätsdruckern – der FUJITSU DL 5600. Eine Hochgeschwindigkeitsmaschine, die 5 Drucker in einem vereint: Durch die unglaubliche Rasanz von ca. 200 Zeilen/Minute ersetzt er jeden Banddrucker. Lagerlisten druckt er ruck-zuck. Korrespondenz erledigt er nicht nur im Düsentempo und Top-Qualität, sondern bietet dabei auch eine große Schriftenvielfalt, die durch IC-Karten noch erweitert wird. Kombinationen von Schrift und Bild druckt er ebenso blitzschnell und sauber aus wie professionelle Farbgraphiken.
Dabei ist der FUJITSU DL 5600 höchst einfach zu bedienen. Durch die 16-stellige LCD-Anzeige wissen Sie immer sofort, was läuft. Durch die 90° Papierzuführung dürfen Sie das Wort Papierstau ein für alle Mal vergessen. Per Knopfdruck schalten Sie problemlos von Endlos- auf Einzelblattbetrieb. Und alles läuft so angenehm leise wie auf Samtpfoten ab.
Ein Hochgeschwindigkeitsdrucker also, für den auch Sie sich blitzartig entscheiden sollten.
FUJITSU DL 5600.
Der professionelle Hoch-
windigkeitsdrucker.
Bitte senden Sie mir genauere Informationen über
mc 2/88
den FUJITSU DL 5600
das gesamte FUJITSU-Drucker-Programm
Name
Straße
PLZ/Ort
Firma
Coupon an: FUJITSU Deutschland GmbH
Rosenheimer Str. 145 · 8000 München 80
Drucker-Telefon-Hotline 089/41301152
FUJITSU
Drucker von Japans Computerhersteller Nr.

BRIEFE

Spaghetti, Spaghetti!

Nachdem in letzter Zeit in Ihrer Zeitschrift eine sehr interessante Diskussion um Programmierstil und strukturierte Programmierung im Gange ist, möchte ich auch ein paar Worte dazu sagen. Besonders einige Veröffentlichungen in Heft 12/87 fand ich sehr anregend.

Dem Beitrag auf Seite 78 von Herrn Reklies kann ich in seiner inhaltlichen Aussage nur zustimmen, wenngleich er auch etwas drastisch formuliert ist. Eigentlich sollte die Verwendung von aussagekräftigen Variablennamen in der Zeit der Compiler und billigen Massen- und Hauptspeicher, in der wir uns heute befinden, eine reine Selbstverständlichkeit sein. In der professionellen Programmierung sind solche Variablennamen ein unverzichtbarer Teil der Programmdokumentation. Das gleiche gilt für Kommentare. Schließlich nimmt ein auch noch so langer Kommentar und ein übersichtlicher und aufgelockerter Quellcode in einem compilierten Programm kein einziges Byte und keine Millisekunde Laufzeit weg.

Was die Verwendung von Sprungbefehlen angeht, so wurden darüber bereits ganze Bücher geschrieben, betrachtet man aber den Leserbrief von Herrn Heisig auf Seite 8, anscheinend leider umsonst. Zwar gebe ich ihm recht, daß bei der Assemblerprogrammierung Sprünge unverzichtbar sind, doch ansonsten hat mich diese Zuschrift eher amüsiert. Abgesehen davon, daß ich der Überzeugung bin, daß Herr Weizenbaum seine Aussagen auch heute jederzeit wiederholen würde und sich damit bei weitem nicht der Lächerlichkeit preisgibt, scheint es mir, daß der Autor dieses Briefes Sinn und Zweck der modernen Programmierung mit mächtigen Kontrollstrukturen, i. A. strukturierte Programmierung genannt, nicht verstanden hat. Abgesehen davon, kann man sogar in Basic strukturiert und ohne GOTOs programmieren.

Zum Artikel von Herrn Reklies möchte ich auch noch sagen, daß sich sein Beispiel bezüglich der (Nicht)Verwendung von GOTOs noch kürzer formulieren läßt unter der Voraussetzung, daß 'bedingung' nicht für alle Schleifendurchläufe getestet werden muß:

```
for (laufbedingung)
  if (! bedingung)
    {
    printf("error");
    return;
    }
printf("weiter");
```

Zum Glück hat ja auch die Mehrzahl der Hobbyprogrammierer zu einer guten strukturierten Programmierung gefunden. Für jemanden, der sich nicht ständig und hauptberuflich mit Programmierung beschäftigt, ist Dokumentation und Erklärung genauso wichtig wie für professionelle Programmierer. Es bleibt nur zu hoffen, daß die Anzahl derjenigen, denen wegen ihrer Grundkenntnisse in Sachen Computer strukturierte Programmierung sauer aufstößt und als sinnlose Zeit- und Speicherplatzverschwendung erscheint (er hat sicher noch nie 100 KB fremdes Spaghettibasic zu debuggen versucht), immer geringer wird.

Dipl.-Ing. Herbert Paulis, A-1050 Wien

Angebot

Ich habe einen Bauvorschlag zur Erweiterung des direkt adressierbaren Speicherbereichs vom Apple-II+ auf 192 KByte bzw. 256 KByte für die mc-68008-Karte (mc 9/85). Ich bin bereit, diesen Vorschlag gegen einen frankierten und adressierten Briefumschlag sowie 40 Pfennig in Briefmarken an Interessenten abzugeben.

Reinhard Ballay
Gerhard-Hauptmann-Weg 12
4352 Herten-Westerholt

MS-DOS mit Komfort

Sehr interessant war Ihr Artikel „MS-DOS mit Komfort“. In 'C' scheint das Problem viel „ungeschwätziger“ programmierbar zu sein (siehe Listing). Zusätzlich habe ich die Möglichkeit eingebaut, Dateien (alternativ zur Kommandozeile) abzuarbeiten. Es sind zwei 'X'-Aufrufarten möglich:

X prog1 arg1 arg2#prog2 arg1 #prog3 arg1 arg2 arg3#.....
X $ textfile.ext oder
X $textfile.ext

In der Textdatei gefundene Zeilenvorschübe werden automatisch als „#“ interpretiert – ansonsten gilt das Format der Kommandozeile. Es werden folgende Funktionen der Lattice-Library verwendet:

printf( )
– formatierte Ausgabe für Usertexte
open( )
– öffne ein File zum Lesen
read( )
– lies eine Anzahl von Zeichen
close( )
– schließe die geöffnete File
strncpy( )
– kopiere Blöcke
strlen( )
– wie viele Zeichen bis Nullterminator?
kbhit( )
– Taste auf der Konsole gedrückt?
system( )
– führe EXEC-Funktion aus

Vielleicht möchten Sie das „Prögrämmchen“ als Leserbrief abdrucken? *Wolfgang Sturm, 2110 Buchholz/i. d. N.*

```
#define FILEREQ '$'                     /* textfile statt kommandos     */
#define TRENNER '#'                     /* der allgemeine trenner       */
#define END     0                       /* der null-terminator          */
#define SPC     ' '                     /* das leerzeichen              */
#define CR      '\r'                    /* das steht in textfiles       */
#define LF      '\n'                    /* oder vielleicht auch das     */
#define BUFLEN  10000                   /* maximale laenge der textfile */
#define MODE    0x8000                  /* files binaer oeffnen         */
                                        /* diverse texte f.benutzer     */
#define TEXT0   "\r\n\007use: X prog1 arg1 arg2#\
prog2#prog3 arg1 arg2#..\r\nor:  X $ textfile"
#define TEXT1   "\r\n'X' finished"
#define TEXT2   "\r\n'X'-request : %s"
#define TEXT3   "\r\n\007command-file: %s not found, 'X' terminated"
#define TEXT4   "\r\n'X' Version V1.0 (c)1987 Ing-Büro Hollmann, Hamburg"

static char buffer[BUFLEN];             /* fuer alle kommandos....      */

main(argc,argv)                         /* mache alles schoen brav      */
register int argc;                      /* wie viele argumente ?        */
register char *argv[];                  /* string-array                 */
{
        register int argp, len;         /* arbeitsvariable              */
        register char *buf, *p;         /* pointer in den buffer        */

        printf(TEXT4);                  /* hello world....              */
        if(argv[1][0] == FILEREQ) {     /* file als kommando-input      */
          if((argp = open(p = !argv[1][1] ? argv[2] : argv[1] + 1,MODE)) < 0) {
            printf(TEXT3,p);            /* fehler-meldung raussenden    */
            return;                     /* fehler-exit                  */
          }                             /* bereite file nun auf         */
          for(buf=buffer;read(argp,buf,1)==1 && buf != buffer + BUFLEN;buf++) {
            if(*buf == CR) *buf = END;  /* ein komplettes kommando      */
            else if(*buf <= LF) buf--;  /* den einfach ueberlesen       */
            else if(*buf == TRENNER) *buf = END;        /* ersetzen     */
          }                             /* der buffer stimmt nun        */
          close(argp);                  /* file noch schliessen         */
        }                               /* bereite kommando-zeile auf   */
        else for(argp=1,buf=buffer;argp < argc;argp++,*(buf = p) = SPC,buf++) {
          strncpy(buf,argv[argp],len = strlen(argv[argp]));
          for(p = buf;p != buf + len;p++) if(*p == TRENNER) *p = END;
        }                               /* nun arbeite befehle ab       */
        if(buf != buffer) for(*--buf = END,p = buffer;(p < buf)&&(!kbhit());
                          printf(TEXT2,p),system(p),p += strlen(p) + 1);
        printf(argv[1][0] != FILEREQ && argc == 1 ? TEXT0 : TEXT1);
        return;                         /* ende-text und fertig         */
}

/* ENDE */
```



ZENTRALE
Florenz-Sartorius-Str. 2
3400 Göttingen
☎ 05 51 / 6 20 47-49
Tx 965 202
Fax 05 51 / 6 20 40

NIEDERLASSUNG ESSEN
Abteistr. 10
4300 Essen-Werden
☎ 02 01 / 49 19 79
Neu



Erweiterungen für XT

- 2. Laufwerk 245,00 DM
- 14" Flat-Screen Monitor (Bernstein o. S/W), 22 MHz, 295,00 DM
- 20 MB Festplatte incl. Contr. 615,00 DM
- Speichererweiterung auf 640 KB 140,00 DM
- Tastatur m. separatem Nummern- und Cursoblock (101 Tasten) 49,00 DM

8 MHz Turbo-XT

- Gehäuse in AT-Ausführung mit Reset- und Schlüsselschalter
- LED-Anzeige für Power und Festplatte
- 8088-2 CPU, (8087 Option)
- 150 W Netzteil
- 640 KB Mainboard (256 KB RAM best.)
- Turbogeschwindigkeit 4,77/8 MHz
- 360 KB Floppy-Laufwerk (Made in Japan)
- Mono-Grafikkarte (Hercules)
- Multi I/O Karte: 2 × RS 232 (1×best.), 1 × par. Printer, Gameport, Clock, Kalender
- DIN-Tastatur 84 Tasten
- Betriebssystem DOS 3.1
- Deutsches Handbuch

885,00 DM*

10 MHz Turbo-XT

wie 8 MHz Turbo-XT jedoch
- Taktgeber 4,77/10 MHz

935,00 DM*

12 MHz Turbo-XT

wie 8 MHz Turbo-XT jedoch
- V20 CPU ● Taktgeber 4,77/12 MHz

1.033 DM*





Der neue Standard!

14" Flat-Screen Monitor
295,00 DM

Alle Systeme **jetzt** mit Flat-Screen Monitor lieferbar!



10 MHz Profi-AT

Norton 10.3

- Gehäuse wie IBM AT, ausbaufähig für alle Plattenlaufwerke, Slimline und hohe Bauart, z.B. 40-100 MB
- Schlüsselschalter für Tastatur
- Taktfrequenzschalter ● Reset-Taste
- LED-Betriebs-, Turbo- und Festplattenanzeiger
- CPU 80286 (80287 Option)
- umschaltbar 6/10 MHz
- Mainboard aufrüstbar auf 1 MB
- 8 Slots (6 AT+2 XT) ● 512 KB RAM best.
- 1 x 1,2 MB Diskettenlaufwerk
- Mono-Grafik/Printer-Karte (Hercules)
- Batteriegep. Uhr/Kalender
- Parallele Schnittstelle
- 200 Watt Netzteil ● DIN Tastatur 84 Tasten
- 14" Flat-Screen Monitor (Bernstein o. S/W) Aufpreis 295,00 DM
- Aufpreis für Tastatur mit separatem Nummern- u. Cursorblock 49,00 DM
- Speichererweiterung auf 640 KB 67,00 DM, auf 1 MB 155,00 DM
- Aufpreis f. 2. Laufw. 1,2 MB 325,00 DM
- Aufpreis f. 2. Laufw. 360 KB 279,00 DM
- Aufpreis f. serielle Schnittstelle 59,00 DM
- Aufpreis für AT I/O-Karte (Gameport, Printerport, 2 serielle Ports) 155,00 DM
- Aufpreis f. 20/40 MB Festplatte m. Controller 925,00 DM/1.299,00 DM
- Betriebssystem DOS 3.1
- Deutsches Handbuch

1.795,00 DM*

12 MHz Profi-AT

Norton 13.3

wie 10 MHZ Profi-AT jedoch
- Taktgeber 6/12 MHz

1.855,00 DM*

12 MHz Kompakt-AT

Norton 13.3

- Gehäuse für 3 Slimline Laufwerke oder 2 FDD + 1 HDD mit Schlüssel-, Reset- und Turboschalter. Gehäusehöhe ausreichend für alle Standard-AT-Karten
- CPU 80286 (80287 Option)
- umschaltbar 6/12 MHz
- Mainboard aufrüstbar auf 1 MB
- 8 Slots (6 AT + 2 XT)
- 512 KB RAM bestückt
- 1 x 1,2 MB Diskettenlaufwerk
- Mono-Grafik/Printer-Karte (Hercules)
- Batteriegepufferte Uhr/Kalender
- Parallele Schnittstelle
- 200 Watt Netzteil ● DIN Tastatur 84 Tasten
- 14" Flat-Screen Monitor (Bernstein o. S/W) Aufpreis 295,00 DM
- Aufpreis für Tastatur mit separatem Nummern- u. Cursorblock 49,00 DM
- Speichererweiterung auf 640 KB 67,00 DM, auf 1 MB 155,00 DM
- Aufpreis f. 2. Laufw. 1,2 MB 325,00 DM
- Aufpreis f. 2. Laufw. 360 KB 299,00 DM
- Aufpreis f. serielle Schnittstelle 59,00 DM
- Aufpreis für AT I/O-Karte (Gameport, Printerport, 2 serielle Ports) 155,00 DM
- Aufpreis f. 20/40 MB Festplatte m. Controller 925,00 DM/1.299,00 DM
- Betriebssystem DOS 3.1
- Deutsches Handbuch

1.795,00 DM*

Alle Geräte sind nach den gültigen Bestimmungen der Bundespost funkentstört.

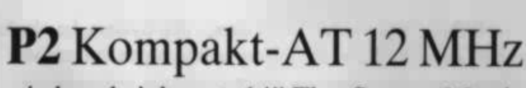
* Preise ohne Monitor, jedoch mit Tastatur (84 Tasten). Aufpreis für erweiterte Multifunktionstastatur: DM 49,00 DM

PAKETPREISE

P1 Turbo-XT 12 MHz

wie beschrieben + 14" Flat-Screen Monitor + erweiterte Tastatur (101 Tasten)

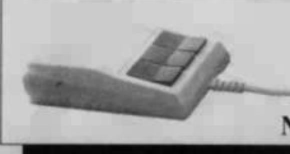
1.329,00 DM



P2 Kompakt-AT 12 MHz

wie beschrieben + 14" Flat-Screen Monitor + erweiterte Tastatur (101 Tasten)

2.115,00 DM



Genius GM-6 Maus für IBM
Microsoft-Kompatibel
95,00 DM
Neu! GM-6 Plus 139,00 DM

EGA-Set

EGA Monitor und Super EGA Karte

Emuliert jeden Standardgrafikmodus (Hercules, EGA, CGA etc.) bis zu 800 × 600 Punkten

1.345,00 DM

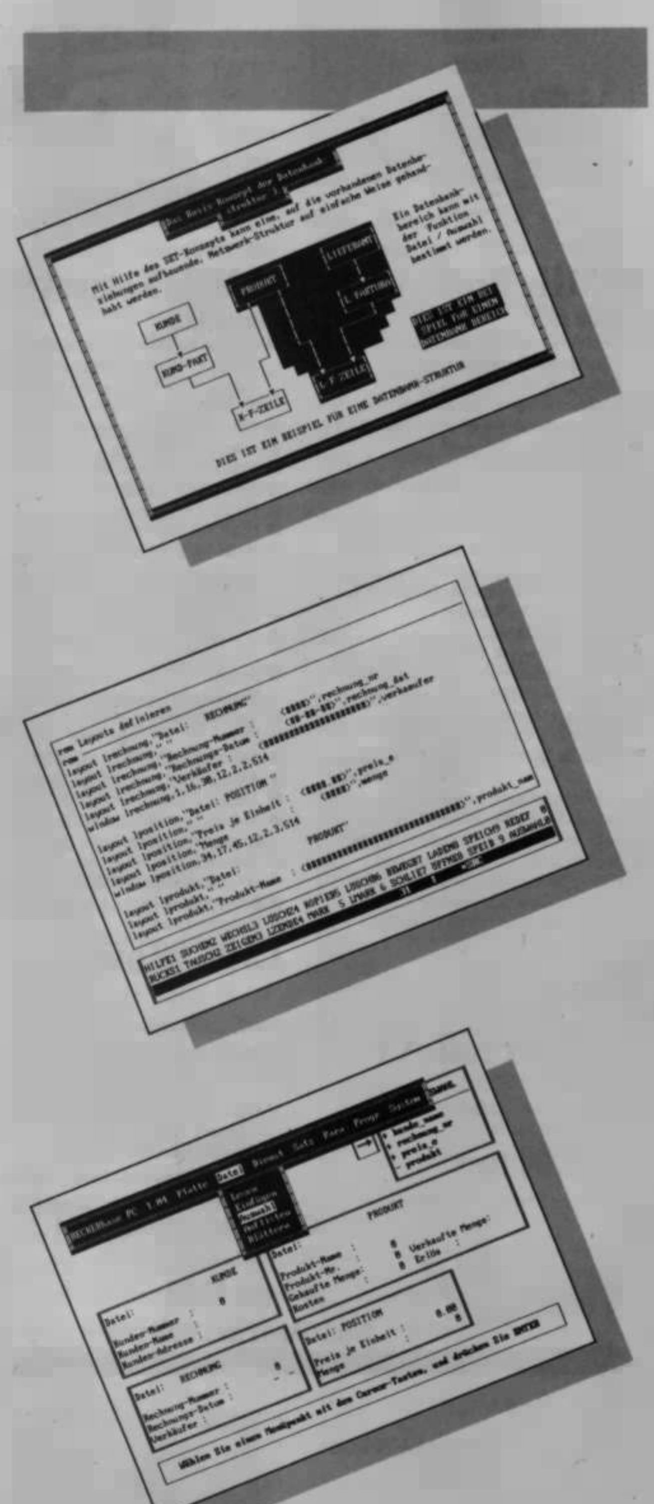


• 1 Jahr Garantie • Technische Betreuung • Eigener Reparatur-Service •

mc 2/88



DATA BECKER INFO



Eine Datenbank setzt sich durch

BECKERBASE PC – ein Datenbank-Manager, der sich nicht an die Standards relationaler Datenbanken hält. Denn BECKERbase PC setzt neue Maßstäbe – als Netzwerk-Modell mit vielen modernen Features.

Mehrdateien-Netzwerk-Struktur

Anders als bei hierarchisch aufgebauten Datenbanksystemen sind die Beziehungen zwischen den einzelnen Dateien netzförmig angelegt. Dadurch lassen sich alle gespeicherten Informationen beliebig miteinander verknüpfen.

Zwei integrierte Programmiersprachen

Hohe Flexibilität in der Anpassung auf spezielle Benutzerbedürfnisse – garantiert durch die leistungsfähige, integrierte Datenbank-Sprache TDL, flexibel durch die Datenbeschreibungssprache DDL. BECKERbase bietet damit alle Möglichkeiten, eigene Anwendungskonzepte zu realisieren.

Kommunikationsfähig

Problemloser Datenaustausch mit anderen Programmen durch spezielle Software-Schnittstellen (ASCII-Format). Zeitaufwendige Neueingaben entfallen.

Mehrere Demo-Anwendungen

Auch ohne Kenntnis der Programmiersprachen kann der Anwender mit den Demo-Anwendungen arbeiten (Adreß-, Artikel-, Kundenverwaltung und andere mehr).

Besondere Features

Pull-Down-/Window-Technik, integrierter Text-Editor, Hilfe-System, komfortables Installationsprogramm, einfache Datei-Definition, unbegrenzte Anzahl von Datensätzen pro Datenbank, 65 535 Datensätze pro Datei, unbegrenzte Anzahl von Feldern je Datei, maximale Feldgröße 255 Zeichen. Minimalkonfiguration: 256 KByte frei verfügbarer Speicherplatz und zwei Diskettenlaufwerke.

Starker Preis

BECKERbase PC, der flexible Datenbank-Manager, setzt neue Maßstäbe. In der Leistung und im Preis. Einfach überzeugend.

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BECKERbase PC



ERT ZUM THEMA PC



In diesem Buch findet der Fortgeschrittene das Detailwissen, mit dem er das Betriebssystem optimal für die tägliche Arbeit nutzen kann. Ob zur Batch-Datei, zum Debugger oder zur Konfigurationsdatei – keine Frage, die unbeantwortet bleibt. Die DOS-Versionen bis 3.3 sind berücksichtigt und werden mit allen Kommandos, Syntax, Erklärungen und Beispielen aufgeführt.

Das große Buch zu MS-DOS
Hardcover, 427 Seiten
DM 49,–

Das gesammelte Wissen zum PC. Hier finden Sie alles zum Systemaufbau eines PC, XT oder AT. Recherchiert und zusammengetragen von einem erfahrenen Software-Entwickler, der weiß, welche Informationen Sie wirklich brauchen. Zum Beispiel bei der Programmierung Ihres Systems in BASIC, C, Turbo Pascal oder Assembler.

PC Intern
Hardcover, 767 Seiten
DM 69,–

Sie werden staunen, was Sie aus Ihrem PC noch alles herausholen können. Verwenden Sie die bewährten Kniffe erfahrener Programmierer. Aus dem Inhalt: neue Tastaturbelegung, 3-D-Grafiken, Notizenverwaltung, batteriegepuffertes RAM, Installation von GEM, Mausprogrammierung und vieles andere mehr.

PC Tips & Tricks
Hardcover, 395 Seiten
DM 49,–

Ziehen Sie alle Register Ihres PCs. Das große Maschinensprachenbuch macht's möglich. Anhand von Beispielprogrammen finden Sie den Einstieg in die Geheimnisse der Maschinensprache: die Unterschiede zu den Hochsprachen, der Makroassembler MASM, Interrupt-Technik, die unterschiedlichen Prozessoren und, und, und. Dazu spezielle Leckerbissen, wie das Einbinden von Assemblerprogrammen in BASIC, Turbo Pascal und C.

Das große Maschinensprachebuch zum PC
Hardcover, 600 Seiten
DM 49,–

Führer zu Norton Utilities. Mit Hinweisen zur Version 3, 4 und der Advanced Edition.

DATA BECKER Führer zu Norton Utilities
Hardcover, 186 Seiten
DM 29,80



Nicht nur, daß die Programme wesentlich schneller laufen, auch die eigentliche Programmierung ist mit Turbo BASIC leichter. Wie einfach, das zeigt Ihnen das vorliegende Buch: Schleifen, Datentypen, Unterprogramme, Programmaufbau, Diskettenbetrieb, Dateiverwaltung…

Das große Buch zu Turbo-BASIC
Harcover, 441 Seiten
DM 49,–

Turbo C wird die Standard-Programmiersprache – eine Einschätzung, die weit verbreitet ist. Halten Sie den Anschluß: Mit diesem Buch sind Sie mit dabei. Hier wird alles einsteigergerecht serviert: Operationen, Argumente, Strings, Schleifen, Zeiger, Felder und Strukturen. Zahlreiche Tips und Tricks sowie eine Übersicht über die häufigsten Programmierfehler sichern die erfolgreiche Handhabung.

Turbo C für Einsteiger
Hardcover, 316 Seiten
DM 49,–

Für alle, die in Turbo Pascal programmieren wollen, wird hier das komplette Know-how geliefert: praxisnahe Einführung, Grundlagen von MS-DOS/PC-DOS, Tips zur rationellen Programmerstellung bis hin zur Dokumentation. Ergänzend dazu Programmprojekte, eine Programmsammlung, die auch als Baustein-Bibliothek verwendet werden kann.

Das große Buch zu Turbo Pascal
Hardcover, 654 Seiten
DM 59,–

COUPON

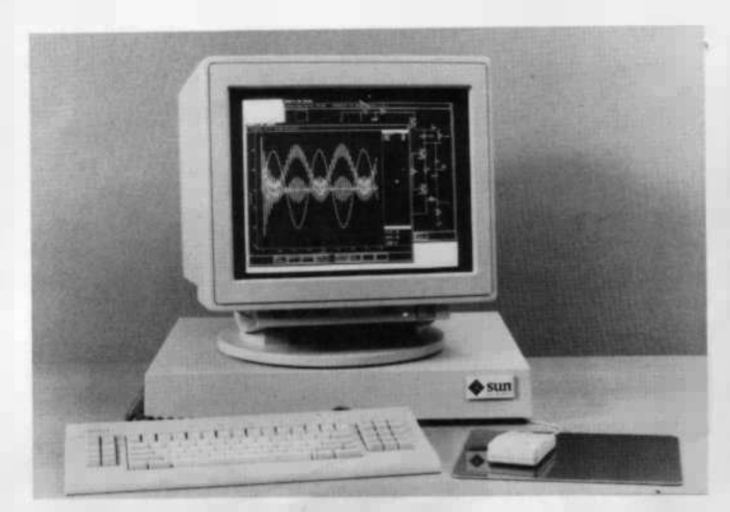
HIERMIT BESTELLE ICH

NAME, VORNAME

STRASSE, ORT

mc 2/88

zzgl. DM 5,– Versandkosten unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei



Der Analog-Entwurfs-Editor auf der Cadnetix CDX 9600

Cadnetix kooperiert mit Sun

Cadnetix, führender Hersteller von CAE-/CAD-/CAM-Ausrüstungen für den Entwurf elektronischer Systeme kündigte jetzt mit Concept 3 ein Spektrum von Systemen auf der Basis von Sun-Workstations an. Die Vereinbarungen zwischen Cadnetix und Sun haben einen Umfang von 27 Millionen DM. Die neuen Cadnetix-Systeme beginnen mit der Sun 3/60er Workstation. Zu einer Leistung von 3 MIPS bieten die Cadnetix-Systeme einen 20-MHz-Gleitkomma-Prozessor, einen hochauflösenden 16-Zoll-Farbmonitor, 4 MByte RAM (auf 24 MByte erweiterbar), einen Ethernet-Transceiver, ein 60-MByte-Kassetten-Laufwerk und eine 140-MByte-Festplatte an. Zu den CAD-Systemen kommt ein Grafikprozessor mit einer Leistung von 400 000 Vektoren in der Sekunde, die für das computergestützte Layout von Leiterplatten erforderlich sind. ○

Neue Hilfsmittel für den Techniker-Arbeitsplatz

Der Computer ist für den Techniker und Ingenieur ein unerläßliches und vertrautes Werkzeug zur Durchführung umfangreicher Rechnungen mit Zahlen geworden. Nach langjährigen Vorarbeiten in vielen Forschungsinstitutionen in Europa und den USA gibt es nun Computerprogramme, die das Rechnen mit Formeln (d. h. mit symbolischen Ausdrücken) gestatten. Diese Programmsysteme bedeuten eine Revolution für das technisch/naturwissenschaftliche Rechnen und erhöhen den Grad der Computerunterstützung der Arbeit des Technikers, und damit seine Produktivität, drastisch.

Wegen der begrenzten Verfügbarkeit auf Großrechnern und der hohen Kosten war die Verwendung solcher Systeme lange Zeit auf einen sehr kleinen Personenkreis beschränkt. An der Universität von Hawaii wurde nun ein Paket für Mikrocomputer (IBM PC und Kompatible) entwickelt (muMATH). Damit besteht nun erstmals die Möglichkeit des breiten Einsatzes dieser Technologie an den Arbeitsplätzen der Techniker und Ingenieure.

Neue Technologien verlangen meist neue Entwicklungsmethoden. Im Bereich des symbolischen Rechnens hat sich, wie auch in vielen anderen neuen Bereichen, LISP als die geeignetste Programmiersprache erwiesen. Es zeichnet sich heute schon ab, daß LISP zukünftig in der Technik eine ähnliche Rolle spielen wird, wie FORTRAN heute. Hand in Hand mit der Entwicklung von muMATH ging auch die Entwicklung von „muLISP“ einher, einem sehr effizienten LISP-Dialekt für PCs.

INFO

Das „Research Institute for Symbolic Computation“ (RISC-LINZ, Johannes Kepler Universität, Linz, Österreich. Ansprechpartner: Dipl.-Ing. B. Kutzler) beschäftigt sich seit Jahren mit diesen Themen in Forschung und Ausbildung und hat sich im Sinne eines schnellen Technologietransfers bereit erklärt, als europäischer Ansprechpartner für muMATH und muLISP zu fungieren. Eine rasche Nutzbarmachung dieser neuen Methode der Computerunterstützung ist von grundlegender Bedeutung für die Erhaltung der Konkurrenzfähigkeit und könnte der europäischen Industrie einen entscheidenden Wettbewerbsvorteil gegenüber der amerikanischen und japanischen Konkurrenz bringen. Noch viel mehr als die Einführung des Taschenrechners (als Hilfsmittel zur Durchführung numerischer Rechnungen) wird die Einführung dieser Methoden (als Hilfsmittel zur Durchführung analytischer (symbolischer) Rechnungen) die tägliche Arbeit eines jeden Ingenieurs und Technikers erleichtern und ihm gestatten, sich auf die wesentlichen, kreativen Teile seiner Tätigkeit zu konzentrieren. ○

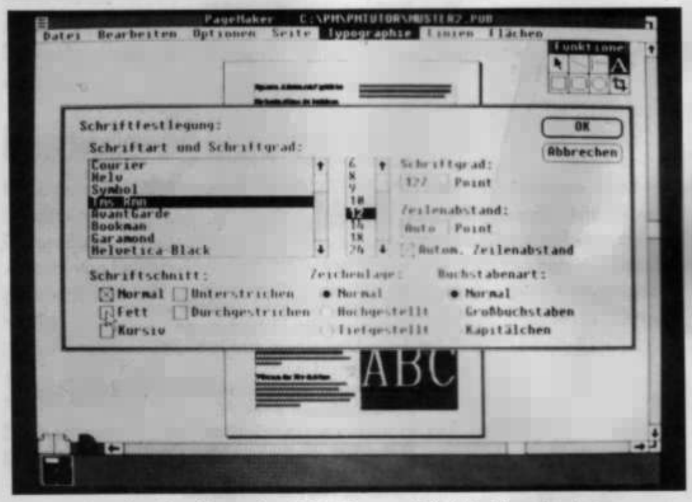
Personal Publishing System von IBM

Ein schlüsselfertiges System für Desktop Publishing hat IBM vorgestellt, das aus dem Modell 30 der PS/2-Serie, einem Monochrom- oder Farbbildschirm, der IBM-Maus, dem Seitendrucker 4216 und der Postscript-Adapterkarte besteht. Postscript ist eine Seitenbeschreibungssprache, die die Ausgabedaten des Dokumentengestaltungs-Programms Pagemaker in Steuerbefehle für Drucker oder Satzmaschinen umwandelt. Im Gegensatz zu vielen anderen Systemen erfolgt die Umsetzung nicht im Drucker selbst, sondern über eine Adapterkarte im PS/2 bzw. PC. Der Vorteil der hohen Geschwindigkeit des internen Datenverkehrs macht sich in einer kürzeren Bearbeitungszeit bei umfangreichen Dokumenten bemerkbar.

Mit einer Video-Schnittstelle werden die formatierten Seiten mit einer Datenübertragungsrate von 1,8 MBit/s von der Postscript-Adapterkarte zum Drukker übertragen.

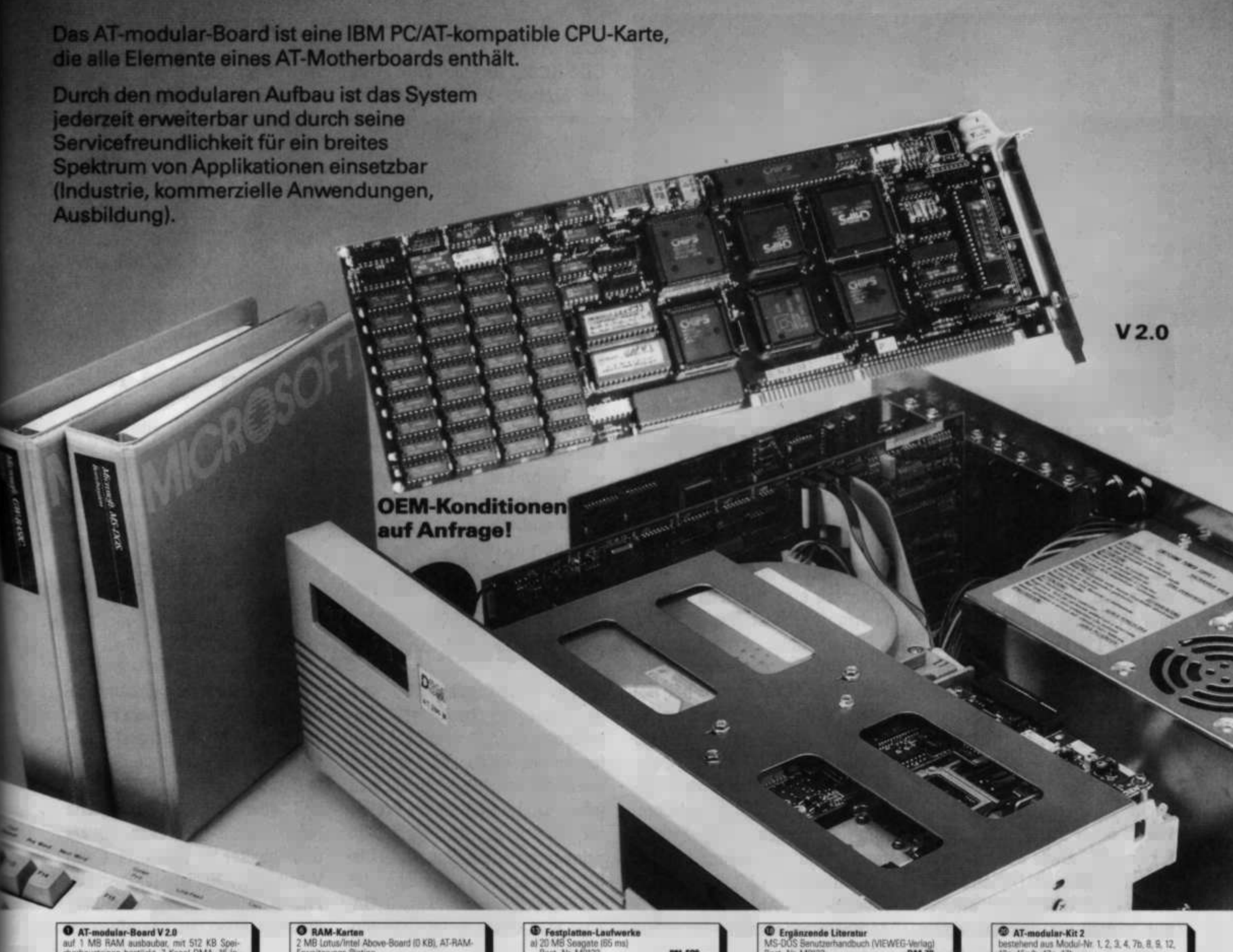
Der Laserdrucker IBM 4216 kann Grafiken, Bilder und Texte mit einer Auflösung von 300 Bildpunkten pro Zoll wiedergeben. Die maximale Druckgeschwindigkeit beträgt sechs Seiten pro Minute.

Wesentliche Bestandteile des Personal Publishing Systems sind die grafische Benutzeroberfläche Windows und das Dokumentengestaltungs-Programm Pagemaker. ○



Pagemaker ist der Kern des IBM Personal Publishing Systems

DER INDUSTRIESTANDARD
DSM-modular-AT



1 AT-modular-Board V 2.0
auf 1 MB RAM ausbaubar, mit 512 KB Speicherbausteinen bestückt, 7 Kanal DMA, 16 Interrupt-Ebenen, 3 programmierbare Timer, 1 Realtime-Clock, CMOS-RAM, bis 64 KB ROM, Keyboard-Controller, 10 MHz, 0-Wait! Neuer CHIPS-TECHNOLOGY-Satz und mit Standard-Speicherbausteinen bestückbar. Inkl. Handbuch
Best.-Nr. M0102, 100 ns RAMs **DM 1098,–**
Best.-Nr. M0100, 120 ns RAMs **DM 998,–**

2 Bus-Platine
mit insgesamt 8 Steckplätzen, davon 6 AT-Slots sowie 2 XT-Slots
Best.-Nr. M0103 **DM 98,–**

3 Festplatten-Floppy-Controller
Verwaltet bis zu 4 Laufwerke, 2 Floppy-Laufwerke (360 KB / 1.2 MB) und 2 Hard-Disk-Laufwerke. Optional 2 zusätzliche Floppy-Laufwerke
Best.-Nr. M0129 **DM 398,–**

4 Monochrom-Grafik-Karte
HERCULES™-kompatibel, mit parallelem Druckeranschluß
Best.-Nr. M0107 **DM 148,–**

5 EGA-Standard-Grafik-Karten
EGA 640 × 350, 640 × 200, 320 × 200, CGA, MDA, 16/64 Farben
Best.-Nr. M0108 **DM 398,–**
SUPER-EGA Level 3 640 × 480, 640 × 350, 640 × 200, 320 × 200, 16/64 Farben
Best.-Nr. M0110 **a. A.**
SUPER-EGA Level 4 640 × 528, 640 × 480, 640 × 350, 640 × 200, 320 × 200, 16/64 Farben
Best.-Nr. M0109 **DM 498,–**
SUPER-EGA Level 5 800 × 600, 640 × 480, 640 × 350, 640 × 200, 320 × 200, 16/64 Farben
Best.-Nr. M0111 **DM 798,–**

* Technische Änderungen vorbehalten

6 RAM-Karten
2 MB Lotus/Intel Above-Board (0 KB), AT-RAM-Erweiterungs-Platine
Best.-Nr. M0112 **DM 398,–**
bestückt mit 2 MB RAM
Best.-Nr. M0113 **DM 998,–**
2.5 MB Multifunktionskarte (0 KB)
AT-RAM-Erweiterungs-Platine mit paralleler und serieller Schnittstelle
Best.-Nr. M0114 **DM 398,–**
bestückt mit 3 MB RAM
Best.-Nr. M0115 **DM 1298,–**

7 Gehäuse
a) mit Befestigungsmaterial
Best.-Nr. M0121 **DM 159,–**
b) stabilere Ausführung, mit Befestigungsmat.
Best.-Nr. M0122 **DM 198,–**

8 Netzteil
200 W kurzschlußfest incl. Netzkabel
Best.-Nr. M0128 **DM 238,–**

9 Tastatur
MF-2-kompatibel, 102 Tasten
Best.-Nr. M0126 **DM 298,–**

10 3½" 1.44 MB Floppy-Disk-Laufwerk
IBM-PS/2-kompatibel (2 MB unform.)
Best.-Nr. M0104 **DM 398,–**

11 3½" 1.2 MB Floppy-Disk-Laufwerk
IBM-AT-kompatibel (1.6 MB unform.)
Hersteller: TEAC / NEC
Best.-Nr. M0105 **DM 298,–**

12 5¼" 1.2 MB Floppy-Disk-Laufwerk
IBM-AT-kompatibel (1.6 MB unform.)
Hersteller: TEAC / NEC
Best.-Nr. M0106 **DM 298,–**

13 Festplatten-Laufwerke
a) 20 MB Seagate (65 ms)
Best.-Nr. M0123 **DM 598,–**
b) 40 MB Seagate
Best.-Nr. M0124 **DM 1150,–**
c) 80 MB Seagate (28 ms)
Best.-Nr. M0125 **DM 3150,–**

14 Coprozessoren
80287 (6 MHz)
Best.-Nr. M0116 **DM 428,–**
80287-8 (8 MHz), für AT-modular
Best.-Nr. M0117 **DM 598,–**
80287-10 (10 MHz)
Best.-Nr. M0118 **DM 740,–**

15 Monitore
14" Monitor mit Schwenkfuß
a) grün
Best.-Nr. M0134 **DM 325,–**
b) bernstein
Best.-Nr. M0135 **DM 329,–**
c) **EGA-Farb-Monitor**
Auflösung bis 640 × 350 Punkte
Best.-Nr. M0136 **DM 998,–**
d) **EIZO-Farb-Monitor**
Auflösung bis 820×620 Punkte für den Betrieb aller EGA-Karten, TTL-Eingang, umschaltbar auf ANALOG-Eingang
Best.-Nr. M0137 **DM 1850,–**

16 Betriebssystem
MS-DOS 3.2 deutsch mit GW-BASIC und deutschen Handbüchern
Best.-Nr. M0120 **DM 298,–**

17 Zubehör
a) Handbuch AT-modular-Computer im praktischen Kunststoffordner
Best.-Nr. M0130 **DM 29,–**
b) Kabelsatz (2 × Floppy und 1 × Hard-Disk)
Best.-Nr. M0131 **DM 49,–**

18 Ergänzende Literatur
MS-DOS Benutzerhandbuch (VIEWEG-Verlag)
Best.-Nr. M0132 **DM 78,–**
IBM PC AT
Rechner der nächsten Generation
(McGraw-Hill-Verlag)
Best.-Nr. M0133 **DM 48,–**

19 AT-modular-Kit 1
bestehend aus Modul-Nr. 1, 2, 3, 4, 7b, 8, 9, 12, 17a, 17b
Best.-Nr. M0201 **DM 2698,–**

20 AT-modular-Kit 2
bestehend aus Modul-Nr. 1, 2, 3, 4, 7b, 8, 9, 12, 13a, 15a/b, 17a, 17b
Best.-Nr. M0202 **DM 3650,–**

21 AT-modular-Fertiggerät
wie Pos. 20, zusätzlich MSDOS 3.2 und GW-Basic komplett aufgebaut und geprüft (48 Stunden Burn-In), 1 Jahr Garantie
Best.-Nr. M0203 **DM 3998,–**

Bei den hier aufgeführten Karten handelt es sich um qualitativ hochwertige Module, die auch in herkömmlichen IBM- oder IBM-kompatiblen Systemen einsetzbar sind. Auf alle Produkte ½ Jahr Garantie!
IBM ist eingetragenes Warenzeichen von der IBM Corp.

Fordern Sie unsere ausführlichen Listen (PC-News und PC-Software-News mit über 500 Programmen) an!

Händleranfragen erwünscht. (Bitte mit Händlernachweis)

DSM
Digital Service GmbH

Zentrale: Landwehrstr. 37 · 8000 München 2
Filiale: Schillerstr. 17 · 8000 München 2

Unsere Hotline für Sie ☎ 089-596916

Telex 523545 dsm d · Fax 089/598261

DSM
ComputerSysteme
Ein Unternehmensbereich der DSM *Digital* Service GmbH



Auch bei LC-Displays hat jetzt das Farbzeitalter Einzug gehalten

LCD-Vorführsystem in Farbe

Telex Communications zieht einen Schlußstrich unter das graue Zeitalter monochromer Vorführsysteme. Magnabyte 5220-I präsentiert Grafiken, Tabellen und andere PC-Inhalte in fünf verschiedenen Farbnuancen. Das einfach zu handhabende und preisgünstige System besteht aus einer auf den meisten Overhead-Projektoren leicht zu befestigenden LCD-Platte, einer Schnittstellenkarte für den PC sowie einer Fernbedienung. Die LCD-Platte und die Schnittstellenkarte müssen nur verbunden werden, schon wird das PC-Bild auf der Leinwand farbig abgebildet, ohne daß dazu eine spezielle Software erforderlich ist. Das IBM XT-/AT-kompatible Vorführsystem ist bei den intelligenten Workstations von Telex und bei fast allen IBM-kompatiblen Anlagen einsetzbar und eignet sich für monochrome und farbige Computer- Systeme. Der Einsatz bietet sich an in Schulen, bei Computerlehrgängen, in Lehrinstituten und Kommunaleinrichtungen sowie in Marketing- und Management-Abteilungen größerer Unternehmen. ○

CeBIT 1988 vom 16. bis zum 23. März

Rund 2300 Aussteller aus 40 Ländern zeigen Ende März auf der CeBIT in Hannover auf 320000 qm Ausstellungsfläche die neuesten Entwicklungen, Trends, Forschungsergebnisse und Highlights aus allen Bereichen der Büro-, Informations- und Telekommunikation sowie der C-Techniken. Die CeBIT 1988 steht ganz im Zeichen der Anwender. Speziell den Problemen und Erfordernissen der mittelständischen Unternehmer widmet sich das Anwendungszentrum Mittelstand und Freie Berufe. Branchen- und aufgabenspezifische Komplettlösungen stehen hier im Vordergrund. Vom kommerziellen Einsatz der gesamten Computertechnologie und der optimalen Software über die technischen Anwendungen bis hin zu CIM ist das Angebot eine „runde Sache". Weitere Schwerpunkte und Sonderveranstaltungen widmen sich den Bereichen



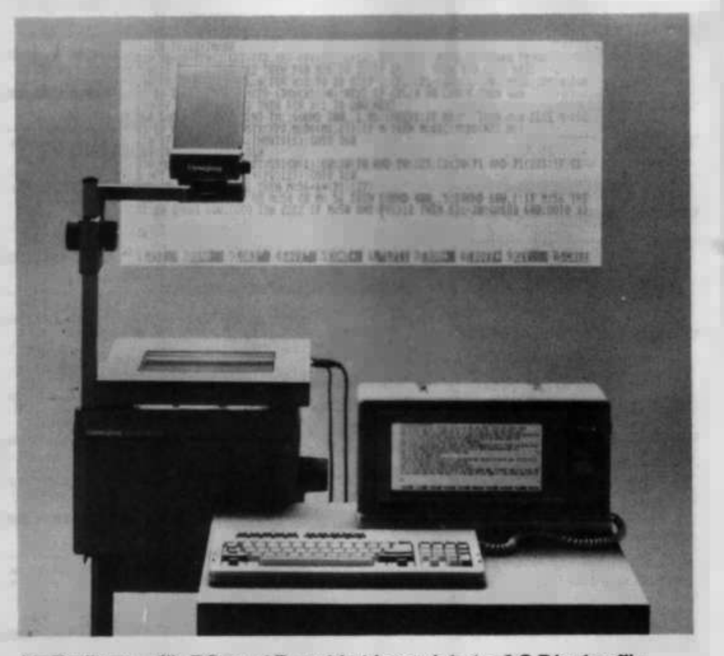
Motto der CeBIT '88: Der Computer läßt den Anwender nicht im Regen stehen

- MultiNET, Demonstration einer offenen, systemunabhängigen Vernetzung
- Telekommunikation, Digitalisierung und Integration
- Das sichere Rechenzentrum, Sonderschau Musterlösungen vom Brandschutz bis zur Datensicherung
- Vom Electronic Banking zur künstlichen Intelligenz
- C-Techniken, Beschleunigte Realisierung von CIM-Lösungen durch einheitliche Schnittstellenprotokolle
- Kommunales Anwender-Centrum, Anwender beraten Anwender
- CeBIT Forum '88, Treffpunkt für Mittelstand und Freie Berufe
- Büro- und Informationssysteme
- Büro- und Organistionstechnik
- Software und Unternehmensberatung
- DV-Peripherie-Geräte
- Zahlreiche Fachtagungen und Kongresse

Der Katalog steht ab Februar zur Verfügung.
Wie in jedem Jahr gibt es Sonderflüge aus dem In- und Ausland, Sonderzüge direkt bis zum Messegelände und Hubschraubertaxis zwischen Flughafen und Messegelände. ○

CGA-Grafik projizieren

Das Data-Display 200 von Liesegang projiziert Computerdaten direkt vom PC-Monitor auf eine Bildwand. Als Lichtquelle für das Display eignet sich jeder DIN-A4-Overhead-Projektor, der nach dem Durchlichtprinzip arbeitet. Das Data-Display wird an die RGB-Buchse des PC angeschlossen und gibt CGA- oder EGA-Grafiken im CGA-Modus mit 640 × 200 Punkten wieder. Die Wiedergabe von Farben kann mit drei Farbsignal-Schaltern beeinflußt werden. Das Data-Display projiziert die Farben hell oder dunkel. Zusätzliche Funktionen wie Freeze, Invers und Schalter für Feinjustage von Vertikal-, Horizontal-Position, Phase und Frequenz erlauben, die Projektion zu beeinflussen.



Als Ergänzung für PCs und Portable bietet sich das LC-Display für Overhead-Projektoren an


Der NL-10 von Star läßt sich über Module an die meisten Computer anschließen

Unix-Mehrplatzgrafik bei Siemens

In seinen Unix-Systemen bietet Siemens nun auch Mehrplatzgrafik an. Damit können die Vorteile von Grafik-Arbeitsplätzen wie einfache Bedienung und größerer Funktionsumfang generell an allen Arbeitsplätzen eines Sinix-(Unix)-Mehrplatzsystems genutzt werden. Die objektorientierte Bedieneroberfläche und die Fenstertechnik erleichtern die Arbeit am Bildschirm erheblich. Dank der Fenstertechnik können mehrere Bildschirminhalte gleichzeitig dargestellt werden. Der gleichzeitige Ablauf mehrerer Programme kann ebenso in mehreren Fenstern verfolgt werden. Der Grafik-Bildschirmarbeitsplatz 97808 ist mit einem eigenen Prozessor und Speicher ausgerüstet und kann sowohl in der Grafik-Betriebsart als auch in der alphanumerischen Betriebsart arbeiten. Der SW-Bildschirm mit 15 Zoll Diagonale hat eine Auflösung von 792 × 594 Pixel, die einzeln ansteuerbar sind. Eine Bildwiederholfrequenz von 70 Hz sorgt für eine flimmerfreie Darstellung. Der Preis für einen Grafik-Bildschirm-Arbeitsplatz liegt bei etwa 6840 DM. ○


Für die Sinix-Systeme von Siemens gibt es jetzt eine Mehrplatzgrafik

Star-Drucker als Drucker-Star

Star Micronics meldete einen Verkaufsrekord: Über 650 000 Drucker vom Typ NL-10 wurden seit der Vorstellung auf der CeBIT 1986 verkauft. Damit ist der NL-10 der bislang erfolgreichste 9-Nadel-Drucker. Mit nur 795 DM ist er ideal für den EDV-Einsteiger. Mit seinem 9-Nadel-Druckkopf erreicht er eine Schönschreib-Druckmatrix von 23 × 18 Punkten. Fünf KByte Speicher, halbautomatischer Einzelblatteinzug und Schubtraktor sind serienmäßig. Interface-Steckmodule ermöglichen den Anschluß an fast alle PCs.

Neben diesem Renner bietet Star eine breite Drucker-Palette, die am oberen Ende mit dem „Laser-Printer 8" abschließt. ○

Standard für CD-Write-Once

Auf gemeinsame Spezifikationen für ein „Compact-Disc (CD) Write-Once-System" für professionelle Anwendungen haben sich Philips und Sony geeinigt. Dabei handelt es sich um einen neuen Standard auf der Basis der weltweit eingeführten Compact-Disc, wobei der „CD-Write-Once"-Recorder einmaliges digitales Aufzeichnen von Daten oder Audio-Signalen auf eine leere „Write-Once"-Disc ermöglicht. Der ursprüngliche CD-Standard wurde um zusätzliche Spezifikationen, z. B. Spur- und Tracking-Signale, erweitert. Ein Daten-Recorder dieses Systems wird auch konventionelle CD-ROM-Discs abspielen können, während auf einem professionellen CD-Write-Once Audio-Recorder auch DC-Audio Discs abgespielt werden können. ○

Schreib-/Lese-Geschwindigkeit:	1,2 bis 1,4 m/s
Spurbreite:	1,6 µm
Disc:	einseitig bespielbar
Disc-Abmessungen:	Durchmesser 12 cm Dicke 1,2 mm
Speicherkapazität:	600 MByte 60 Minuten audio
Modulation:	EFM-Code
Fehler-Korrektur:	Cross Interleave Reed-Solomon Code (CIRC)

Technische Daten des CD-Write-Once-Systems

Bestseller Wordperfect

Mit 38 000 verkauften Programmen in knapp 15 Monaten und einem Marktanteil von etwa 18 Prozent hat sich Wordperfect einen Spitzenplatz bei den Textprogrammen in Deutschland erkämpft. In den USA steht das Programm mit über 800 000 verkauften Exemplaren und einem Marktanteil von 38 Prozent noch besser da.

Der Umsatz der deutschen Wordperfect GmbH aus Eschborn liegt derzeit bei etwa 7 Millionen DM und soll sich 1990 auf 30 Millionen DM steigern. Neben dem Textverarbeitungsprogramm bietet Wordperfect jetzt auch ein Tabellenkalkulationsprogramm (Planperfect) und ein Datenbankprogramm (Dataperfect) an. Über die einheitliche Benutzeroberfläche Library werden die speicherresidenten Programme miteinander verbunden. Die Library bietet sechs Programme: Rechner, Kalender, Dateiverwaltung, Notizbuch, Makro- und Programm-Editor und ein Spiel. Das Programm Dataperfect belegt nur 128 KByte RAM bei einer Speicherkapazität von 17 GByte. Für 1988 hat Wordperfect ein Desktop-Publishing-Programm angekündigt. Mit einem Zusatzprogramm von Mikado aus Berlin kann man mit Wordperfect jetzt auch avestische, georgische und armenische Texte schreiben. Wordperfect ist außer für MS-DOS-PCs auch für Laptops, Amiga und Atari ST verfügbar. ○

Das Haus für Industriecomputersysteme



Σ-X86-19

Der Null "Wait State" Rechner Σ-X86-19*: das Industriecomputersystem.

Speziell für den industriellen Einsatz:

- Robust: stabiles Stahlblech-Chassis
- Zuverlässig: MTBF 3500 h
- Störsicher: spezieller Netzfilter, hochwertige Netzteilmodule
- Color- und Monochrombildschirm
- Wahlweise 3½" oder 5¼" Laufwerke
- SCSI-Interface bis 600 MByte
- LAN-fähig

Das Chassis

- AT-kompatibler Industrierechner in 19"-Einschubtechnik für den Einsatz in rauher Fertigungsumgebung
- Industriegerechter, robuster Hardwareaufbau in 19"/4HE Einschubtechnik
- Front- und Rückplatte aus 3 mm Al beidseitig kunststoffbeschichtet mit geschweißten Befestigungs- und Massebolzen
- Seitenteile und Abdeckblech 1,5 mm Stahlblech gelb chromatiert
- Ausschnitte in der Frontplatte für 2×3½" oder 2×5¼" Laufwerke
- Netzschalter und Lüftungsschlitze
- Schlüsselschalter für Tastaturverriegelung
- Anwendungsgerecht konfigurierbar
- Lieferbar mit kundenspezifischem Firmenzeichen

* Steht für 80386/16 MHz und 80286/10 MHz



Spezial-Electronic KG

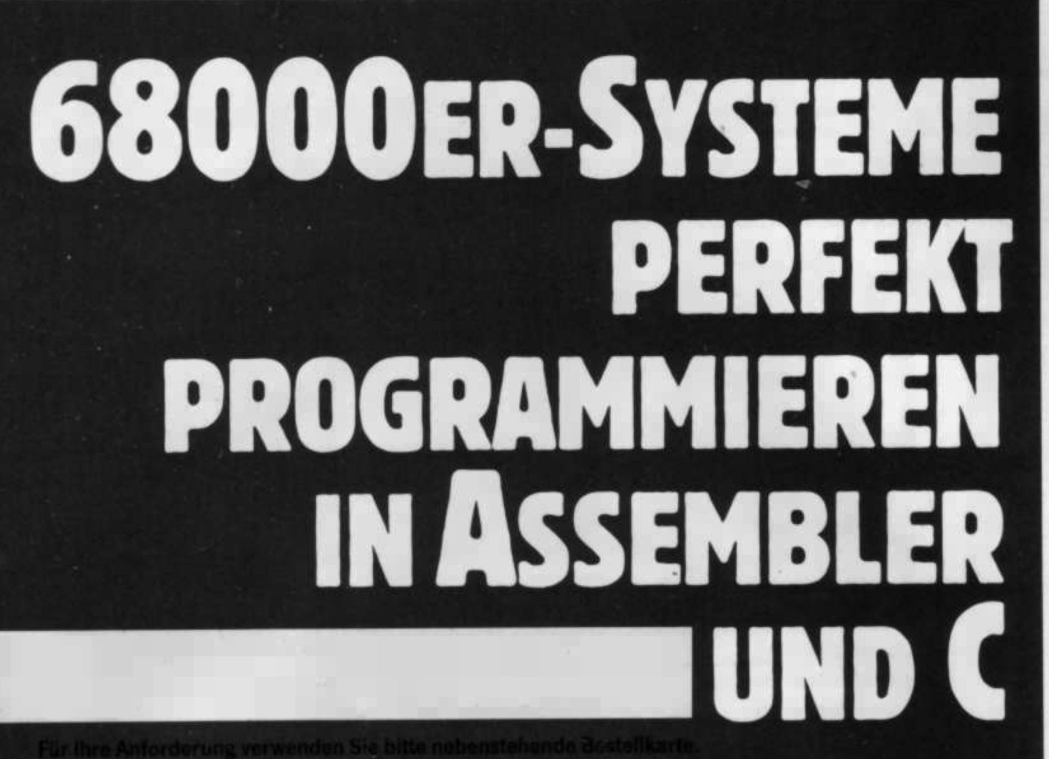
3062 Bückeburg, Postfach 1308
Telefon 05722/203106, Telex 17572210
Teletex 572210, Telefax 05722/203120

6057 Dietzenbach, Max-Planck-Str. 6–10
Telefon 06074/24061, Telex 4191576

7090 Ellwangen, Postfach 1320
Telefon 07961/4047, Telex 17796110
Teletex 796110, Telefax 07961/6030

8000 München 82, Postfach 82609
Telefon 089/429333-338, Telex 5212176
Teletex 898493, Telefax 089/428137

Für alle Orte der BRD zum Ortstarif: 01307367



Für Ihre Anforderung verwenden Sie bitte nebenstehende Bestellkarte.

Erfolgreiches Programmieren von 68000er Systemen in Assembler und C

durch
- detaillierte Baustein- und Betriebssystembeschreibungen
- Trainingsprogramme für perfekte Programmierung
- programmierte Unterweisung in Assembler und C
- mit Bibliotheken und Programmsammlung



Erst prüfen, dann kaufen
Schauen Sie sich dieses Werk in Ruhe an: 10 Tage lang dürfen Sie Ihr Ansichtsexemplar unverbindlich zu Hause prüfen.

Ab sofort hilft Ihnen dieses neue Nachschlagewerk,
- effiziente Problemlösungen auf **Betriebssystemebene** zu realisieren
- anspruchsvolle **Anwenderprogramme** zu entwickeln
- mit ausgefeilter **Bausteinprogrammierung** neue Anwendungsbereiche zu erschließen.

Die wichtigsten Themen auf einen Blick:
- **detaillierte Hardware-Beschreibungen** der Prozessorfamilie 68000 und ihrer 8-, 16-, 32-Bit-Peripheriebausteine mit technischen Kennwerten, Pinbelegung, internen Architekturen, Befehlsbeschreibungen; Zudem erhalten Sie Hinweise für optimale Kombinationen wie zwischen MMC 68461/CPU 68012 und Weiterentwicklungen;
- **Betriebssysteme:** Sie lernen die Strukturen der 68000er-Betriebssysteme (wie GEM-TOS, OS-9 und Unix) kennen und erfahren, wie Sie die Systemroutinen zur Optimierung Ihrer Assembler- und C-Programme nutzen;
- **Softwareengineering:** die optimale Vorgehensweise von der Problemanalyse über die Codierung bis hin zur abschließenden Dokumentation;
- **Programmierkurse für effiziente Anwender-, System- und Bausteinprogrammierung in Assembler und C.** Damit verfügen Sie zugleich über sofort einsetzbare Routinen, z. B. zur Drucker- und Bildschirmansteuerung oder GEM-TOS-Utilities;
- **die C-Compiler-Werkstatt:** Schritt für Schritt programmieren Sie selbst einen C-Compiler inkl. Funktionsbibliotheken (Ein-/Ausgabe, Statistik/Mathematik...);
- **Assembler-Makrobibliotheken** (Stringverarbeitung...), **Anwender- und Funktionsbibliotheken in C** (Dateiverwaltung u. a.);
- **Tips und Utilities** wie Schnittstellen- und Backup-Routinen, Fensterverwaltung;
- **bewährte Musterlösungen** wie zu Rechnerkopplung und Multitasking;
- **Anleitung für raffiniertes Systemtuning,** z. B. durch Einsatz schnellerer Peripheriebausteine/Höhertakten der CPU;
- **Praxiswissen zu speziellen Einsatzbereichen** wie Messen, Steuern, Regeln. Mit den ausführlichen Anleitungen erstellen Sie z. B. ein universelles MSR-System mit eigener Peripherie oder programmieren EPROMS;

Dieses Werk veraltet nie
Unsere 68000er-Fachredaktion versorgt Sie regelmäßig und zuverlässig mit aktuellen Informationen über Hardware- und Betriebssystem-Weiterentwicklungen, weiteren Utilities und Musterlösungen sowie Bibliothekserweiterungen in Assembler und C.

Fordern Sie noch heute an:

Erfolgreiches Programmieren von 68000er-Systemen in Assembler und C

stabiler Ringbuchordner, Format DIN A 4, Grundwerk ca. 400 Seiten. Bestell-Nr. 3400, Preis: DM 92,– (erscheint ca. 1. Quartal 1988)

Alle 2–3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 120 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).

INTEREST-VERLAG

Industriestraße 21
D-8901 Kissing
Tel. 08233/23900

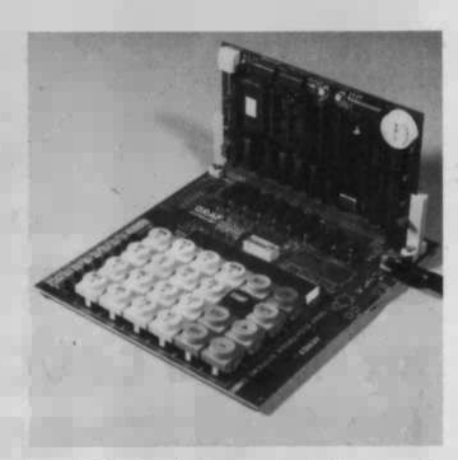
Fachverlag für anspruchsvolle Freizeitgestaltung

GRAF® computer



Graf computer zur Ausbildung

Der NDR-Computer: Ein ideales Werkzeug zur Ausbildung in der Schule, im Betrieb oder im Selbststudium. Modulares Rechnersystem vom Einsteigerpaket (Bild) über Z80 / CP/M 2.2 bis zur 68000-Familie mit 68020. Fast beliebiger Ein- und Ausstieg möglich. Umfangreiche Baugruppen zum Messen / Steuern / Regeln. Unterstützung durch eigene Kundenzeitschrift "LOOP". Gesamtkatalog (DM 10.- Schutzgebühr) anfordern.
Der NDR-Computer wird als Ausbildungscomputer bei der Deutschen Bundespost eingesetzt (M-I-K-Projekt)



Das NDR-Einsteigerpaket: Eingesetzt bei der Deutschen Bundespost

Graf computer: AT-kompatibel

Der mc-modular-AT: Das modulare Konzept auch beim IBM-Kompatiblen, das sich durchgesetzt hat. CPU-Karte steckbar, deshalb auch später gegen -386-Karte auszuwechseln. Kompakter Aufbau, Einsatz in 19" problemlos möglich. Festplatten-Floppy-Controller steuert zwei Platten und bis zu drei Disk-Laufwerke, 5 1/4" oder 3 1/2", bis zu 1.4 MByte (PS/2-kompatibel). Der mc-modular-AT ist als Komplettgerät oder in einzelnen Baugruppen lieferbar. Kostenlosen Prospekt anfordern. Diese Anzeige wurde auf einem mc-modular-AT mit PageMaker gestaltet.

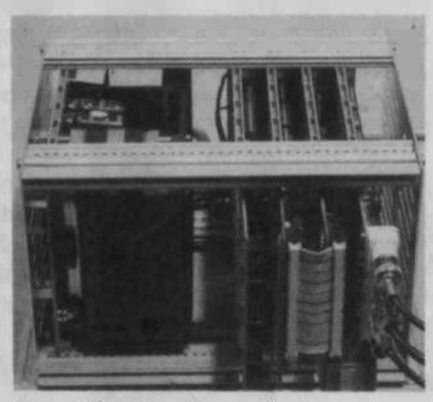


Der mc-modulare -AT
Das schnelle , modulare System.

Graf computer für die Industrie

Graphik-Subsysteme für industrielle Anwendungen:Wir liefern Geräte mit eigenem Prozessor (936x und ACRTC) zur Text- und Graphik-Ausgabe im Maschinenbau, S/W und Farbe.

SPS-Steuerung: Der neue SPS-Compiler erzeugt aus SPS-Code 8080/8085-Assembler Quellcode, der übersetzt auf preiswerten Ein-Platinen-Computern ablauffähig ist. Näheres in unser LOOP16 - mit nebenstehender Bestellkarte kostenlos anfordern.
Wir stellen aus: Hobby-Tronic Dortmund (3.-7.2.) Halle 6 / 6022



Graphik-Subsysteme für den industriellen Einsatz

OEM-Anfragen an:

GRAF ELEKTRONIK SYSTEME GMBH

Magnusstraße 13 · Postfach 1610 · 8960 Kempten (Allgäu) · Telefon: (0831) 6211
Teletex: 831804=GRAF · Telex: 17831804=GRAF · Datentelefon: (0831) 69330

Filiale Hamburg:
Ehrenbergstraße 56
2000 Hamburg 50
Telefon (0 40) 38 81 51

Filiale München:
Georgenstraße 61
8000 München 40
Telefon (0 89) 2 71 58 58

Händler:
Jörg Korb · 1000 Berlin 33 · Telefon (0 30) 8 21 19 47
GMCP mbH · 2800 Bremen 33 · Telefon (04 21) 21 10 88
Schweiz: SYSTECH · CH-4104 Oberwil · Tel. (0 61) 30 10 31



Ein Decoder-Modul wertet die Heimcomputer C-64 und C-128 zur Btx-Station auf

Btx-Modul für C-64/C-128

Jeder C-64 oder C-128 verwandelt sich in eine Btx-Station, wenn man das von Siemens entwickelte Btx-Decoder-Modul aufsteckt. Gefertigt wird das Btx-Modul von Commodore. Der Heimcomputer bringt nicht nur die farbigen Bildschirmtextseiten auf den Monitor, sondern speichert auch mehrere Seiten und fragt Btx-Dienste automatisch ab. Bei dem niedrigen Preis des Zusatzgerätes (derzeit 399 DM) lohnt sich die Anschaffung eines der beiden Heimcomputer auch, wenn das System nur für Btx verwendet wird. Nur acht Mark betragen die monatlichen Kosten für einen Btx-Anschluß.

Auf der Funkausstellung 1987 in Berlin zeigte sich der Bundespostminister Dr. Christian Schwarz-Schilling sichtlich optimistisch beim Anblick des neuen Btx-Moduls. Wenn sich nur ein Bruchteil der 1,5 Millionen Familien, in deren Haushalt ein C-64 oder C-128 steht, das Modul zulegt, braucht sich die Post keine Sorgen mehr um Btx zu machen. ○

Kommunikations-Stadt der Zukunft

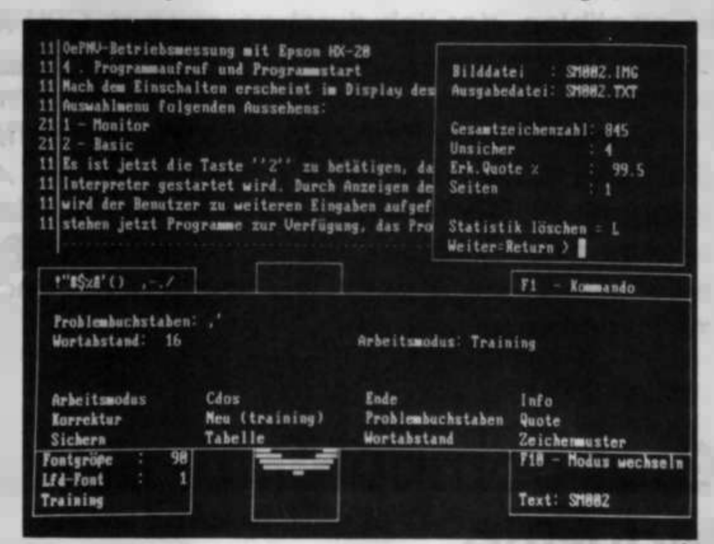
Zum 1. Januar 1988 wird die Forschungsgruppe „Verteilte Systeme" als „Forschungsgruppe für Kommunikations-Systeme" (FOKUS) in die Gesellschaft für Mathematik und Datenverarbeitung eingegliedert. FOKUS wird ein Forschungszentrum der GMD mit dem gleichen Status wie das Berliner „Forschungszentrum für Innovative Rechner-Systeme und technologie" (FIRST). FIRST und FOKUS arbeiten in zwei verschiedenen Bereichen gemeinsam daran, daß die Bundesrepublik auf dem zukunftsentscheidenden Gebiet der Informations- und Kommunikationstechniken auch weiterhin eine führende Rolle spielen kann.
In FIRST entwerfen rund 60 Wissenschaftler und ebensoviele Studenten eine völlig neue Computer-Generation und entwickeln sie bis zur Marktreife. Denn die heutigen Rechner, die stur einen Befehl nach dem anderen abarbeiten, haben die Grenzen ihrer Leistungsfähigkeit erreicht. Künftige Computer müssen komplizierte Aufgaben in Wissenschaft, Forschung und Industrie in Teilprobleme zerlegen, diese von mehreren hundert „Unter-Rechnern" parallel und gleichzeitig lösen lassen und aus den Resultaten das Endergebnis bilden. Dies erfordert neben einem völlig anderen Aufbau auch grundlegend neue Rechenverfahren und Computerprogramme. Mit seinem STARLET-Rechner hat das Team bereits einen international stark beachteten Prototyp der 5. Generation entwickelt.

INFO

Zur Zeit arbeitet die Gruppe am Projekt SUPRENUM mit, dem „Superrechner für numerische Anwendungen", dessen Prototyp mit 256 Einzelrechnern gleichzeitig an Teilaspekten einer einzigen Aufgabe arbeiten können soll. Die rund 50 Mitarbeiter von FOKUS arbeiten daran, daß Computer unterschiedlicher Arbeitsweise und Bauart miteinander kommunizieren können. Das FOKUS-Team wird sich federführend mit dem GMD-Leitvorhaben „Offene Anwendungs- und Interkommunikations-Systeme" (OAI) beschäftigen. Dabei geht er vor allem darum, für die Computer der Zukunft Hochleistungsnetzwerke zu entwerfen für eine schnelle und ihren Fähigkeiten entsprechende Kommunikation. Voraussetzung dafür ist eine breitbandige Übertragungsmöglichkeit, die eine Hochgeschwindigkeits-Übermittlung erlaubt. Vorarbeiten werden in Berlin bereits mit dem BERKOM-Netz geleistet, zu dem alle wissenschaftlichen Einrichtungen der Stadt zusammengeschaltet werden. Die Übertragungsgeschwindigkeit von BERKOM wird mit 560 Millionen Bit in der Sekunde an der Grenze des derzeit Machbaren liegen. ○

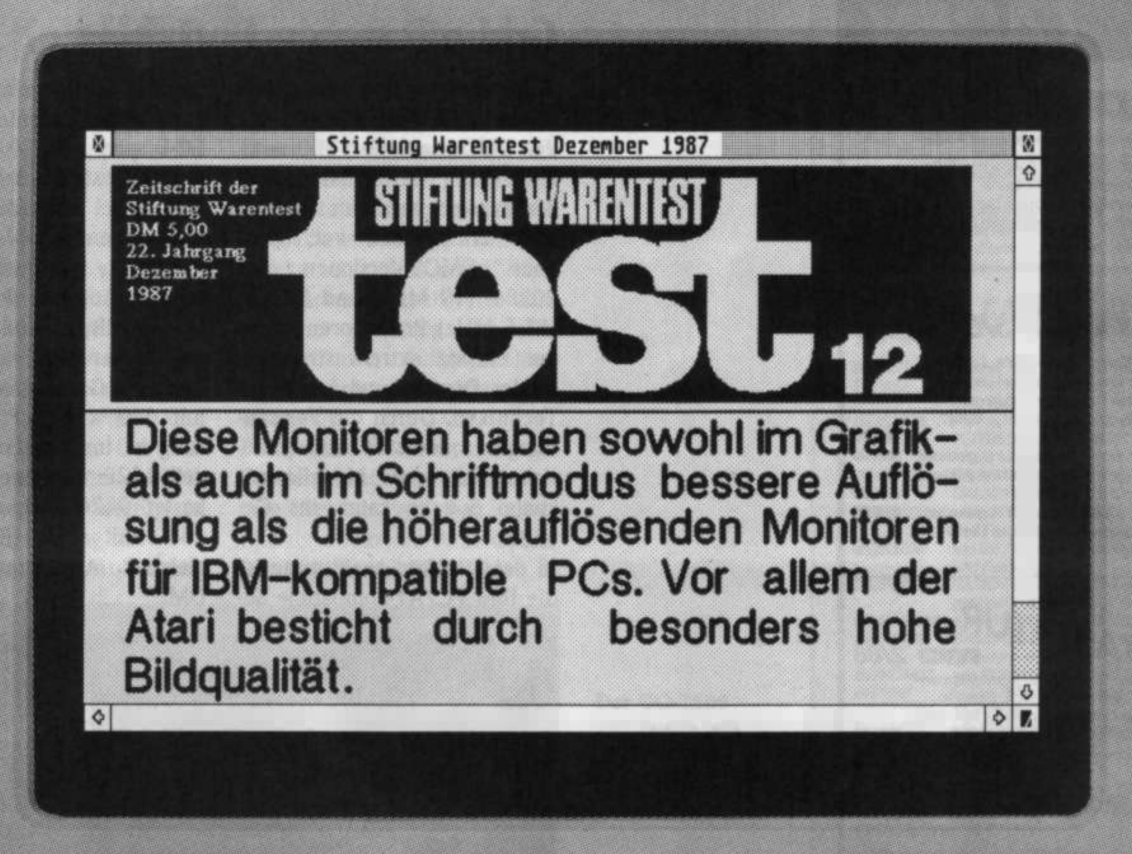
Co-Prozessor erkennt Zeichen

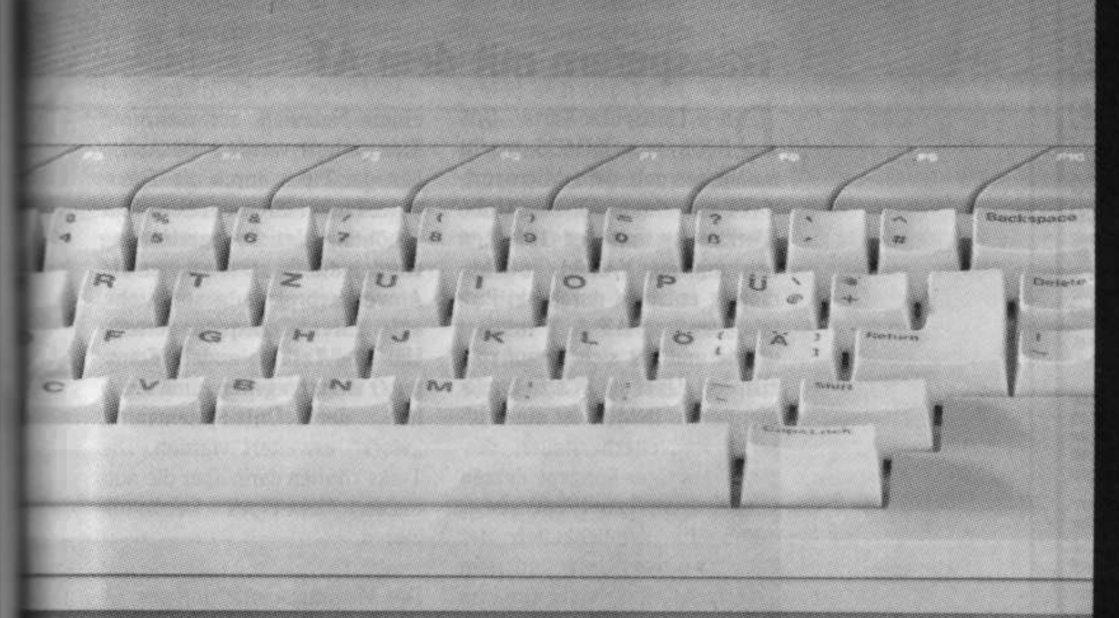
Mit Sesam, einem System zur Erkennung von Schriften und allgemeinen Mustern, stellt Makrolog aus Wiesbaden jetzt ein System vor, das leistungsfähige Zeichenerkennungstechnik und PCs zusammenbringt. Durch eine 32-Bit-Co-Prozessor-Karte (NS32032, 10 MHz) mit mindestens 2 MByte RAM und dem Multitasking-Betriebssystem Concurrent-DOS wird die Leistung eines ATs soweit gesteigert, daß komplizierte Zeichenerkennungs-Algorithmen in einer erträglichen Zeit ablaufen. Das System kann auf beliebige Schriftarten und Zeichenmuster eingerichtet werden. Etwa 1500 Formen werden simultan bearbeitet, wobei etwa 30 bis 50 Seiten in der Stunde bei einer Erkennungssicherheit von 99 Prozent gelesen werden können. Selbst chinesische, kyrillische und griechische Zeichen liest Sesam ein. Außer dem Sesam-Paket, das aus Steckkarte und Software besteht, benötigt man noch einen Scanner. Die eingelesenen Texte liegen als ASCII-Dateien vor. Wer sich über die technische Entwicklung bei der Zeichenerkennung und das gegenwärtige Angebot informieren möchte, kann bei Makrolog eine interessante Broschüre kostenlos abrufen. ○



Sesam hat einen Schreibmaschinentext gelesen und 99,5 Prozent der Zeichen erkannt.

ATARI setzt Maßstäbe – Ihren Augen zuliebe 71 Hz. 640 x 400 Bildpunkte.





Der Monitor ATARI SM 124 hat eine Bildwiederhol-Frequenz von 71 Hz. Das heißt: 71 Mal pro Sekunde wird das Bild wiederholt – das, was Sie auf dem Monitor sehen, sehen Sie also völlig ruhig. Ihre Augen werden nicht gereizt. Folgeerscheinungen wie Ermüdung und Überanstrengung, die zu Fehlleistungen führen, werden vermieden. Der Monitor ATARI SM 124 erfüllt allein damit Voraussetzungen, die von Verbänden und Berufsgenossenschaften als Grundbedingungen gefordert werden. Er setzt Maßstäbe, wie alle ATARI-Geräte der ST-Serie.

Der ATARI SM 124 ist Technologie von heute. Und. Technologie von heute ist preiswerter. Soviel Leistung zu solch' niedrigen Preisen kann Ihnen nur bieten, wer modernste Technologie einsetzt.

ATARI, das ist Computertechnologie für Menschen, die mit mehr Leistung mehr leisten wollen.

ATARI Monitor SM 124 für alle ATARI ST-Computer.



...wir machen Spitzentechnologie preiswert.

Grid mit neuen Portables

Der Laptop-Spezialist Grid aus Düsseldorf stellt mit der GRiDcase-Serie 1500 netzunabhängig betreibbare 286/386-Laptops vor. Den Akkubetrieb ermöglichen vor allem neue CMOS-Versionen der 80286- (10 MHz) und 80386- (12,5 MHz) Prozessoren sowie der Einsatz extrem stromsparender Display- und Laufwerk-Techniken. Durch die Verwendung besonders leichter Materialien fallen die Modelle mit knapp 5,4 kg kaum ins Gewicht.

In der Standard-Ausführung ist die IBM PC/AT-kompatible Serie 1500 mit einem hintergrundbeleuchteten Supertwist-LC-Display von 10 Zoll Diagonale ausgestattet. 1 MByte RAM und zwei 3½-Zoll-Diskettenlaufwerke mit 1,44 MByte sind eingebaut. An Festplatten gibt es eine Auswahl von 10-MByte-, 20-MByte- und 40-MByte-Laufwerken.

Auch ein Gasplasma-Bildschirm mit 640 × 400 Pixeln Auflösung ist im Angebot. Das Modell 1520 mit einem 80C286 kostet 9120 DM, das Modell 1530 mit einem 80C386 bei gleicher Ausstattung 12540 DM. ○



Akkubetriebener 80386-Laptop von Grid mit hintergrundbeleuchtetem LC-Display

Transputern mit dem AT

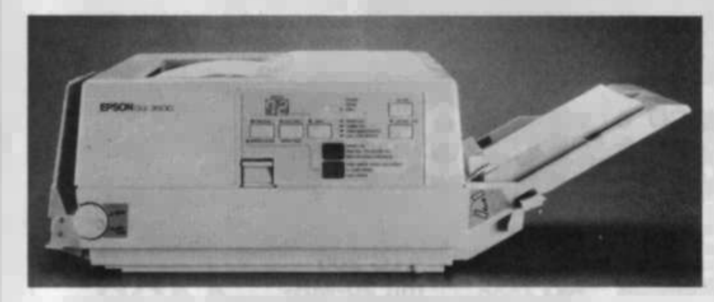
Die Transputer-Karte IMS B008 von INMOS macht zusammen mit dem Microport-V/TT aus einem AT eine Unix-Workstation mit der 15fachen Leistung einer VAX 11/780. Microport entstand durch ein Projekt, das von AT & T initiiert wurde und hat sich darauf spezialisiert, Unix auf PCs zu implementieren. INMOS ist ein britischer Halbleiterhersteller, dessen Transputer-Konzept derzeit Furore macht. Laut INMOS ist es durch die Ähnlichkeiten der Prinzipien von Unix und dem Transputer eine relativ einfache Aufgabe, Anwenderprogramme zu parallelisieren. Ein typisches Unix-Anwenderprogramm besteht aus einer Anzahl von Tasks, die durch Pipes miteinander gekoppelt sind. Über Pipes werden Daten ausgetauscht. In einem Netzwerk aus mehreren Transputern werden die Aufgaben der Pipes durch die Transputer-Links übernommen. Zur Erhöhung des Rechnerleistung werden die Tasks, aus denen ein Anwenderprogramm besteht, auf mehrere Transputer verteilt. Um eine Task an andere Transputer weiterzugeben, muß lediglich der Unix-Systemaufruf „vfork“ erweitert werden. Die Tasks können dann über die normalen Unix-Pipes kommunizieren.

Die Maximalausstattung des Systems V/TT besteht aus einem Motherboard und vier Submodulen, die jeweils mit 2 MByte Speicher und einem Transputer IMS T800 bestückt sind. Ab Anfang 1988 soll das System in Serie gehen. ○

Hoher Komfort – Niedriger Preis

Daß ein niedriger Preis nicht mit Einbußen bei Ausstattung, Leistung, Komfort oder Geschwindigkeit verbunden sein muß, beweist Epson mit dem neuen Laserdrucker GQ-3500. Serienmäßig vorhanden sind Epson Page Printer Mode, HP-Laserjet Plus-Emulation und LQ-1500-Emulation. Bequem über IC-Karten nachladbar sind zusätzliche Schriftarten und weitere Emulationen. Der GQ-3500 wird von einer 68000-CPU gesteuert und verfügt in der Grundausstattung über 640 KByte RAM; eine Aufrüstung auf 1,5 MByte ist vorgesehen. Die Druckleistung beträgt sechs Seiten in der Minute. Der GQ-3500 verfügt über umfangreiche Steuerungsmöglichkeiten am Bedienerfeld und kann Ausdrucke auch „face down" ablegen. Papierzuführung und -ablage sind großzügig dimensioniert, Toner, Auffangbehälter und Trommel sind leicht auszuwechseln.

Kompaktes Äußeres, geringes Gewicht und ein geräuscharmer Betrieb gestatten einen Einsatz direkt am Arbeitsplatz. Der Preis: unter 5700 DM. ○



Der GQ-3500 von Epson soll weniger als 6000 DM kosten

NEC-Multispeed am Nordpol

Einen Kältetest hat der Laptop NEC-Multispeed mit Bravour bestanden. Er begleitete ein Forscherteam auf einer spektakulären Arktis-Expedition. Innerhalb von zwei Monaten wurden Eisuntersuchungen, Messungen der Meeresströmungen und des Salzgehaltes im Wasser, Bestimmungen radioaktiver Spurenstoffe im Meerwasser sowie Messungen zum Ozongehalt in der Stratossphäre vorgenommen. Mit Prüfroutinen wurden die Meßwerte der Sensoren direkt in den Computer eingegeben und noch vor Ort ausgewertet, was zugleich auch die einwandfreie Funktion der Sensoren bestätigte.

Für den NEC-Multispeed entschied sich das Team, weil er wegen seiner robusten Konstruktion als geeignet für Experimente bei Temperaturen um den Gefrierpunkt, hoher Luftfeuchtigkeit und extremer Belastung erschien. Außerdem wurde der Laptop auch als Gesellschafter für einsame Polarabende eingesetzt, z. B. zum Schreiben persönlicher Berichte. ○



Zur Meßwerterfassung diente einer Arktis-Expedition ein NEC-Multispeed



Der PEGA1 ist ein Video-Display-Controller für die Standards EGA, CGA und MDA

Single-Chip für EGA-Videostandard

Mit dem PEGA1 bietet Paradise Systems einen Single-Chip-Videocontroller an, der eine besonders rationelle Implementierung des etablierten EGA-Videostandards erlaubt. Attraktiv an PEGA1 ist, daß auch die übrigen Videonormen (CGA, MDA, Hercules, Plantronics Color-Plus und die Paradise-Farbsimulation mit Grauabstufungen) unterstützt werden. Farbgrafiken werden auf Monochrom-Bildschirmen in 16 Graustufen dargestellt. Scrolling erfolgt im Text- und Grafik-Modus flimmerfrei und ohne Bildrauschen. Ein vollständiges Video-Subsystem benötigt nur zehn weitere Bausteine, zu denen ein EPROM-Zeichengenerator, ein Quarz und Speicherchips gehören. Um ein IBM-PC-kompatibles System zu entwickeln, kann der PEGA1 auf einer separaten Steckkarte untergebracht werden. Zur Minimierung des Platzbedarfs läßt sich der Chip direkt in die Mutterplatine integrieren. Für eine hochauflösende Bildschirmdarstellung kann man PEGA1 auch in ein Terminal einbauen. Der Video-Controllerchip ist einem 84poligen Gehäuse untergebracht. ○

File-Server mit 500 MByte Kapazität

Zur Systems in München stellte die Molecular Computer GmbH aus Eschborn den Hochleistungs-File-Server „Professional System 386-Tower" vor. Das Grundmodell basiert auf einem 80386 der mit 16 MHz getaktet wird. Eine Version mit 20 MHz Taktfrequenz ist ebenfalls verfügbar. An Festplatten stehen Versionen von 80 MByte bis zu 620 MByte zur Auswahl. Mit dem Molecular P/S-386 ist es nun erstmals möglich, unter Novell Advanced Netware betriebene File Server mit über 500 MByte Kapazität in einem Gehäuse zu realisieren. Bei den Festplatten handelt es sich um Hochleistungsplatten mit ESDI-Interface, die eine Datentransferrate von 10 MBit in der Sekunde bei einem Interleave-Faktor von 1 erreichen. Normale ATs erreichen eine Rate von etwa 5 MBit in der Sekunde.

Der File-Server wurde in Zusammenarbeit von Molecular, Schneider & Koch und dem amerikanischen Festplattenhersteller Core-International entwikkelt. Nach Verfügbarkeit des von Novell angekündigten Netware Release 2.1 können File-Server mit bis zu 1240 MByte realisiert werden. ○

Starker Umsatz bei Tandon

In den ersten neun Monaten 1987 hat Tandon den Umsatz um mehr als 50 Prozent auf 259 Millionen Dollar gesteigert und dabei einen Gewinn von 11,7 Millionen Dollar erzielt. Im gleichen Zeitraum 1986 waren es noch 63,6 Millionen Dollar Verlust. Am Umsatzwachstum waren vor allem Computer und Subsysteme mit einem Anteil von zwei Drittel beteiligt, während der Anteil der Laufwerke, mit denen Tandon ins PC-Geschäft einstieg, auf ein Drittel des Gesamtumsatzes fiel. ○

Teamrechner PCD-3T

Der Siemens PCD-3T ist ein Standmodell auf Basis des 32-Bit-Prozessors 80386 (16 MHz, Null Waitstates). Er erweitert die Produktpalette der Siemens PC nach oben hin und verfügt über sechs freie Steckplätze für Erweiterungskarten. Es können bis zu sieben Slimline-Laufwerke eingebaut werden. Integriert werden können Festplatten-, Diskettenlaufwerke und/oder Streamer bzw. auch andere externe Speichermedien. Sowohl 3½-Zoll- als auch 5¼-Zoll-Laufwerke stehen zur Verfügung. Sie können beliebig kombiniert werden. Ab 1988 wird der PCD-3T neben MS-DOS auch mit dem Betriebssystem MS-OS/2 geliefert. Zur Beschleunigung von arithmetischen Rechenoperationen kann der Numerikprozessor eingebaut und die Taktfrequenz auf 6 MHz oder 8 MHz zurückgeschaltet werden. Der Speicherausbau beginnt bei 1 MB und kann auf der Grundplatine bis auf 16 MB erweitert werden. Zum Lieferumfang gehört neben Bildschirm, Tastatur, und Systemeinheit MS-DOS 3.2, MS-Windows mit MS-Write und MS-Paint, GW-Basic und Anwenderhandbuch. Mit einer 40-MB-Festplatte kostet der Rechner 14800 DM, mit einer 70-MB-Festplatte 22000 DM. ○



Der Siemens-PCD-3T

Siemens konzentriert sein PC-Geschäft

Mit Beginn des Jahres 1988 werden die Aktivitäten von Siemens auf dem Gebiet der MS-DOS-/ MS-OS-PCs in Augsburg zusammengefaßt. Bisher waren Entwicklung, Produktion und Vertrieb auf mehrere Standorte in München und Augsburg verteilt. Man geht bei Siemens davon aus, daß durch diese Konzentration und Straffung Entscheidungsprozesse wesentlich beschleunigt werden, und man so auf die Forderungen des schnellebigen PC-Marktes noch intensiver reagieren kann. Künftig soll der Vertrieb über den Fachhandel und Systemhäuser größere Bedeutung bekommen, während sich der Direktvertrieb vor allem um das Großkunden- und Projektgeschäft kümmern wird. Siemens wird wesentliche Anstrengungen auf eine stärkere Internationalisierung des PC-Geschäfts hin unternehmen, um auch in Europa eine bedeutende Position zu erringen.

Der Siemens PCD-2L ist ein Floppy-Gerät (3½ Zoll) auf Basis der bereits beim PCD-2 verwendeten Slot-CPU mit dem Prozessor 80286 und dem Betriebs-System MS-DOS. Es ist für den Einsatz im lokalen Netz gedacht und rundet die Konzeption des Siemens Basisnetzes Comfonet ab. 1988 wird zusätzlich das neue Betriebssystem MS-OS/2 für den PCD-2L angeboten. Der PCD-2L ist voll AT-kompatibel.


Der PCD-2L ist das Einstiegsmodell von Siemens

Ein freier Steckplatz für eine LAN-Anschlußplatine steht zum Ausbau zur Verfügung. Das Gerät verfügt außerdem über eine Centronics- und eine V.24-Schnittstelle. Bildschirm, Tastatur und Systemeinheit entsprechen dem Siemens-Standard. Eine Erweiterungskarte mit drei Betriebsarten übernimmt die Bildschirmsteuerung (trimodal: Monochrom-Adapter-Mode, Color-Graphics-Adapter, Hercules-Graphic-Controller). Der Bildschirm läßt sich beliebig von Schwarz- auf-Weiß in inverse Darstellung umschalten. Auf Wunsch kann auch ein Farbbildschirm mit EGA-Karte angeschlossen werden. Der Rechner wird mit einem 3,5-Zoll-Laufwerk mit 720 KByte angeboten. Zum Lieferumfang gehören MS-Windows mit MS-Write und MS-Paint, GW-Basic, Anwenderhandbuch. Der Preis wird bei 5640 DM liegen. Für den Farbmonitor sind 1500 DM zu zahlen. ○

KOMPATIBEL ... 24-STUNDEN-TEST ... LEISTUNG ... PREIS... QUALITÄT... 1 JAHR GARANTIE

MCI XT 16 SLC

alles drin!

Grundausstattung ohne Monitor

ab 899,–

beinhaltet:

- voll IBM®XT kompatibel
- 8088 CPU + 8087 Sockel
- 8 XT Slots
- 256 KB freier Speicher
- 1 x 360 KB Floppy-Drive
- Color- oder Monochr. Grafikkarte (Hercules II komp. 720 x 348 P.)
- Deutsche Normtastatur MK 5111
- 150 W Schaltnetzteil
- Parallele Drucker-Schnittstelle



Erweiterungen für XT 16 SLC-Serie

2. Laufwerk 360 KB		249,–
Speichererweiterung auf 640 KByte		149,–
Clock/Seriell-Karte		79,–
I/0 Plus II Karte		149,–
20 MB Festplatte mit XT-Controller	+	589,–
30 MB Festplatte m. RLL XT-Contr.	+	679,–
EGA-Set statt monochr. Karte	+	1149,–
Opt. Roll-Maus MO 86 m. Softw.	+	199,–
Professional Multifunktions-Tastatur MK 6000	+	100,–
MS-DOS 3.2 + GW-Basic	+	149,–
9" TTL Monitor grün	+	150,–
12" Monitor grün od. bern.	+	229,–
14" TTL Monitor grün, bern. od. weiß	+	279,–

PRINTER



MCI *Personal Computer Graphics Printer Plus*

- voll kompatibel zum *IBM Personal Computer Graphics Printer*
- 120 Zeichen/sec.

399,–



OKI MICROLINE ML 192 Elite

- 9 Nadel Matrixdrucker
- Druckgeschwindigkeit 200 Z./sec.
- 40 Zeichen/sec. NLQ
- Druckpuffer 8 KB
- IBM Kompatibel

899,–

MCI AT 4 SLC



Grundausstattung ohne Monitor

ab 1899,–

beinhaltet:

- voll IBM® AT kompatibel
- 80286 CPU + 80287 Sockel
- 6 AT + 2 XT Slots
- 8 und **12 MHz** umschaltbar
- 512 KB freier Speicher
- 1 x 1,2 MB/360 KB Laufwerk
- Color- oder Monochr. Grafikkarte (Hercules II komp. 720 x 348 P.)
- Parallele Drucker-Schnittstelle
- Batteriegep. Echtzeituhr/Kalender
- Kapazitive deutsche Normtastatur

NEU 8/12 MHz

Dieses Gerät ist nach den Bestimmungen d. Vfg 1046/84 der Deutschen Bundespost funkentstört

Erweiterungen für AT 4 SLC-Serie

2. Laufwerk 360 KB		299,–
20 MB Festplatte mit AT-Controller		889,–
30 MB Festplatte m. RLL AT-Contr.	+	999,–
Seriell-Karte		79,–
I/0 Plus II Karte		149,–
EGA-Set statt monochr. Karte	+	1149,–
MS-DOS 3.2 + GW-Basic	+	149,–
Professional Multifunktions-Tastatur MK 6000	+	100,–
9" TTL Monitor grün	+	150,–
12" Monitor grün od. bern.	+	229,–
14" TTL Monitor grün, bern./weiß	+	279,–

EGA



Hochauflösender EGA-Monitor

EGA-Karte 249,–

- Auflösung 320 x 200 (CGA Mode) 640 x 350 (EGA Mode)

999,–

TELEFON-HOTLINE-PREISE
MCI Telefonansage (20 sec.)

	(0 22 02)
Festplatten & Controller	10 81 40
Grafikkarten	10 81 41
Monitore	10 81 42
Schnittstellenkarten	10 81 43
Drucker	10 81 44
Aktuelle Neuigkeiten	10 81 45



5060 Bergisch Gladbach 2
Bensberger Straße 252
Tel.-Nr.: 02202/1080
Fax: 02202/31009 · Telex: 8873518

12 Monate Garantie auf alle Geräte. Nach der Pang Vo. v. 14. 3. 85 sind wir bei Angeboten gegenüber dem Endverbraucher zur Angabe der **Preise incl. MwSt.** verpflichtet. Für Druckfehler wird nicht gehaftet. **Preise gültig ab 1. 1. 88.** Lieferzeit und Lieferbedingungen auf Anfrage. Änderungen, die technischen Verbesserungen dienen, vorbehalten. Zwischenverkauf vorbehalten.
MCI MICRO COMPUTER INSTRUMENTS GMBH eingetragen AG Bergisch Gladbach · HRB 2575. Herstellung und Vertrieb von Mikrocomputern. 5060 Bergisch Gladbach 2 · Bensberger Straße 252

Neue Mikroprozessoren von Hitachi

Eine Reihe neuer Hochleistungs-Mikroprozessoren – die sogenannte H-Serie – hat kürzlich Hitachi vorgestellt. In der Prozessorfamilie sind 8-, 16- und 32-Bit-CPUs vertreten. Obwohl Hitachi in der „Ahnenreihe" den Prozessor 68000 aufführt, ist die H-Serie zu ihm nicht Code-kompatibel. Um eine neue Stufe der Leistungsfähigkeit von Mikroprozessoren zu erreichen, hat man alle Brükken zur Vergangenheit abgebrochen und bei der H-Serie neue Konzepte verwirklicht. Der Prozessor H16 ist mit sechzehn Registerbänken mit je sechzehn 32-Bit-Universalregistern ausgestattet. In zwei Betriebsarten können die Registerbänke betrieben werden, dem Global- und dem Ring-Modus. Der Global-Modus ist besonders für Multitasking-Betriebssysteme geeignet, bei denen der Prozessor blitzschnell von einer Task zur nächsten umschalten muß. Konventionelle Prozessoren retten bei der Taskumschaltung, meist Kontextwechsel genannt, die Registerstände auf den Stack. Der H16 kann den Kontextwechsel fast vollständig mit der Registerbankumschaltung durchführen. Im Ring-Modus sind acht der sechzehn Registerbänke ähnlich einem Ring-Puffer geschaltet, die bei Prozeduraufrufen als Cache dienen. Erst wenn alle acht Registerbänke belegt sind, muß der Prozessor auf den Stack zugreifen. Da ein Zugriff auf interne Register immer schneller als ein Zugriff auf externe Register ist, wächst die Ausführungsgeschwindigkeit enorm.
Interessant ist, daß die Hochsprachen C und Modula-2 unterstützt werden. Als Betriebssystem sind Unix und das geheimnisumwitterte ITRON (Industrial Real Time Operating System) vorgesehen. Sogar eine VMEbus-Karte mit dem H16 wurde bereits von Hitachi vorgestellt. ○

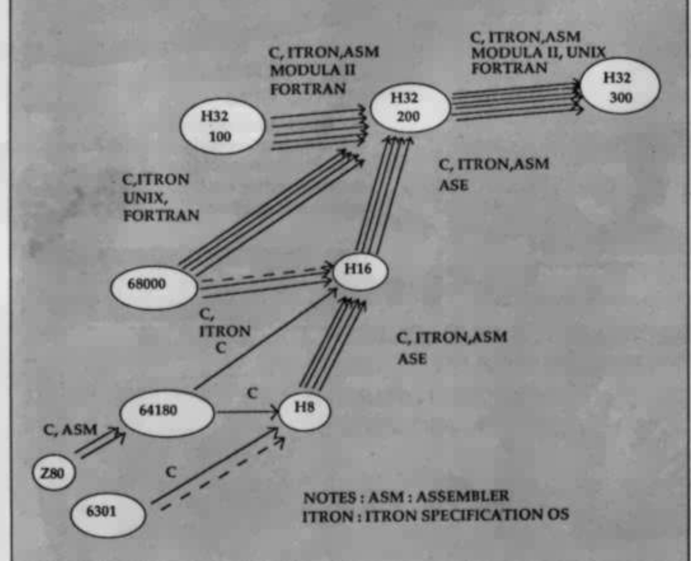


Bild 1. Fünf neue Mikroprozessoren gibt es in der H-Serie

Bild 2. Die H16-CPU hat eine große Zahl von Universalregistern

Stellt sich automatisch auf verschiedene Horizontal- und Vertikalfrequenzen ein: der Monitor 7250 von Conrac

Monitor für alle gängigen Video-Standards

Das Monitormodell 7250 von Conrac synchronisiert sich selbst auf Horizontalfrequenzen zwischen 15 und 37 kHz und auf Vertikalfrequenzen zwischen 47 und 80 Hz. Damit eignet es sich für alle gängigen Grafik-Standards, von CGA bis EGA, PGA und VGA. Wer z. B. bei Textdarstellung keinen Wert auf Farbe legt, kann das Gerät so umschalten, daß die Zeichen nur in Grün zu sehen sind. Die entspiegelte 19-Zoll-Bildröhre liefert auch bei 1024 x 1024 Bildpunkten noch ein gestochen scharfes Bild. TTL- und Analogeingänge vervollständigen die vielfältigen Betriebsmöglichkeiten dieses Monitors. ○

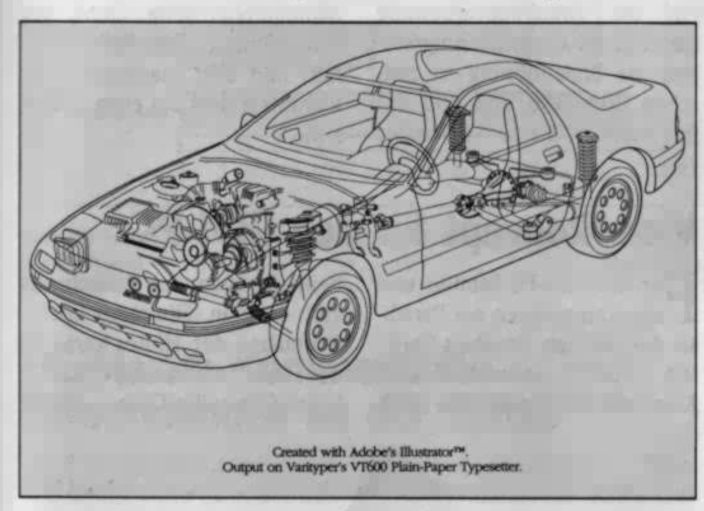
Konstruktion per Telefon

Auf Herz und Nieren ist ein Teil für den Airbus geprüft worden, das am 12. September in Osnabrück konstruiert wurde und nur wenige Minuten später in Bremen fertig einem Fräsautomaten entnommen werden konnte. Obwohl die Prüfzeit über sechs Wochen lang dauerte, ist das „tele-konstruierte" Teil tatsächlich im Flugzeugbau einsetzbar. Initiator dieser Demonstration war Prof. Dr. Dieter Otten von der Uni Osnabrück, der den Teilnehmern eines CIM-Seminars die Leistungsfähigkeit von CAD/CAM zeigen wollte. Während des Seminars wurde das Teil für den Airbus mit PC-Draft von RHV gezeichnet und in ein Steuerprogramm für den NC-Fräsautomaten umgesetzt. Per Datenfernübertragung über das Telefonnetz wurden die Daten an den Rechner in Bremen gesendet. ○

Laserdrucker mit hoher Auflösung

Bisher hatten Laserdrucker eine Auflösung von 300 x 300 Bildpunkten pro Zoll. Für gehobene Ansprüche ist das eine viel zu geringe Auflösung. AM International bietet jetzt den Laserdrucker VT 600 an, der mit einer Auflösung von 600 x 600 Bildpunkten fast schon Fotosatzqualität erreicht. Die hohe Wiedergabequalität wird neben der höheren Auflösung auch durch einen sehr feinen 2-Komponenten-Toner erreicht. Der Durchmesser eines Toner-Körnchens beträgt 7 Mikron gegenüber 13 Mikron bei herkömmlichen Druckern. Der Adobe-Atlas-Bildrechner mit der CPU 68000, der 6 MByte große Speicher und eine 20-MByte-Festplatte machen aus dem Drucker eigentlich einen leistungsfähigen Spezial-Computer.

Neun Schriften sind im ROM gespeichert; vier weitere Schriften werden auf der Festplatte mitgeliefert, die ungefähr 100 Schriften speichern kann. Geeignet ist der Drucker z. B. für Anwender, die hochwertige technische Dokumentationen erstellen müssen. ○



Auch kleinste Details werden mit dem Laserdrucker VT600 gut wiedergegeben

PAL/NTSC-Decoder/Encoder

Für die beiden Normen PAL und NTSC hat Sony Farbsignal-Encoder/Decoder entwickelt. Der Farbsignal-Encoder V 7040 mischt die RGB-Signale aus zwei Quellen (Fernseh- oder PC-Signale) und gibt sie nach Farben getrennt oder als zusammengesetztes Signal an zwei Ausgängen aus, die jeweils 75-Ω-Übertragungsleitungen treiben können. Zur Umwandlung normgemäßer, zusammengesetzter Videosignale in drei analoge Farbauszüge dient der Decoder-Schaltkreis V 7020. Neben den RGB-Signalen liefert dieser Chip Burst-Impulse und Hilfsträger und trennt das zusammengesetzte Synchronsignal ab. Damit sind alle Signale für eine Bildmischung und Verarbeitung bereits aufbereitet. Der Baustein V 7020 ist für die Normen PAL und NTSC verwendbar und über einen Steueranschluß zwischen Farb-Video- und PC-Signalen umschaltbar.

Die Bausteine V 7040 und V 7020 sind beide in 28poligen DIL-Gehäusen erhältlich und benötigen nur eine 5-Volt-Versorgungsspannung.

In Verbindung mit anderen Bausteinen aus dem Sony-Video-Prozessor-Angebot lassen sich wesentlich einfacher als bisher leistungsfähige Systeme zur Digitalisierung und Bildmischung realisieren. ○

Unix auf dem Vormarsch

Im Jahre 1992 werden in der Bundesrepublik voraussichtlich über 100 000 Computer mit dem Betriebssystem Unix ausgeliefert. Diese Prognose stellte das Marktforschungs-Institut IDC aus Eschborn in seiner Studie „Der deutsche Markt für Unix 1986-92" auf. Das Umsatzvolumen wird für 1992 auf 2,8 Mrd. DM geschätzt. 1986 wurden bereits ca. 16 000 Unix-Rechner ausgeliefert. Marktführer im Bereich der Workstations (der etwa 47,8 Prozent ausmacht) sind laut IDC die Unternehmen Hewlett Packard, Sun und Apollo mit 88 Prozent Anteil. Bei kleinen Mehrplatzsystemen (51,5 Prozent) steht Siemens mit über 50 Prozent Anteil an der Spitze vor Altos. Marktführer bei den mittleren Systemen (0,7 Prozent) ist mit über 30 Prozent Anteil DEC. ○



Der Laser-Drucker PT 10 schafft 480 Seiten in der Stunde. Die Druckkosten werden mit 6 Pfennig pro Seite angegeben

Laser-Drucker von Siemens

Der Laser-Drucker PT 10 von Siemens ist ein HP-kompatibler Drucker (Laser Jet Serie II) für etwa 8 200 DM. Für die Layout-Gestaltung bietet er verschiedene Schriftvarianten, -formen und -größen. Hochauflösende Grafiken und Formulare können beliebig gemischt werden. Die Auflösung beträgt 300 × 300 Punkte pro Quadratzoll; die Geschwindigkeit acht Seiten pro Minute bzw. 480 Seiten in der Stunde. Der Speicher von 512 KByte für Grafik- und Font-Memory kann mit steckbaren Platinen von 1 MByte, 2 MByte oder 4 MByte ausgebaut werden. Mit einer 4-MByte-Platine wird aus dem PT 10 ein Multiuser-Ausgabegerät, über das mehrere Benutzer in einem LAN zugreifen können. Der PT 10 verfügt standardmäßig über sechs Schriftarten für die Zeichensätze Roman 8, IBM 8, ISO-7-Bit-Erweiterungen und ECMA 94. Insgesamt faßt der Druckerspeicher 32 nachladbare Schriftarten. Bis zu 16 Schriftkombinationen können gleichzeitig auf einer Druckseite dargestellt werden. ○

Scanner für DTP von Siemens

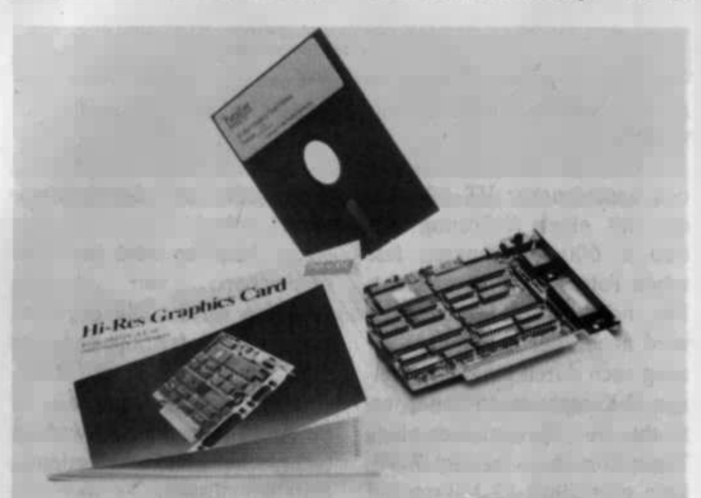
Neu bei Siemens ist der Scanner ST 400, der sich besonders für Desktop Publishing und Archivierungs-Aufgaben eignet. Bedienung und Schnittstellen orientieren sich dabei an internationalen Standards. Mit der Bedieneroberfläche MS-Windows kommt er vor allem für ATs und Kompatible in Frage. Über die SCSI-Schnittstelle kann er auch an jeden anderen Rechner angeschlossen werden. Der Flachbett-Auflage-Scanner ST 400 arbeitet mit Faseroptik und LED-Beleuchtung. Für die Graubildbearbeitung bietet er 64 Quantisierungsstufen; die Bildauflösung ist zwischen 200, 300 und 400 dpi frei wählbar. Bis zu 16 Fenster lassen sich mit verschiedenen Merkmalen versehen. Für eine Seite DIN-A4 in Schwarzweiß braucht der ST 400 in 400 dpi weniger als 10 Sekunden. Neben der MS-Windows sind weitere Versionen der Scanner-Software geplant, z. B. für die Bedieneroberfläche GEM, für Unix (Sinix), VMS (DEC-Rechner) und dem Macintosh. Der Richtpreis liegt bei etwa 10260 DM. ○



Mit einer maximalen Auflösung von 15,85 Punkten pro Millimeter arbeitet der Scanner von Siemens

Videokonzept für den IBM-PC

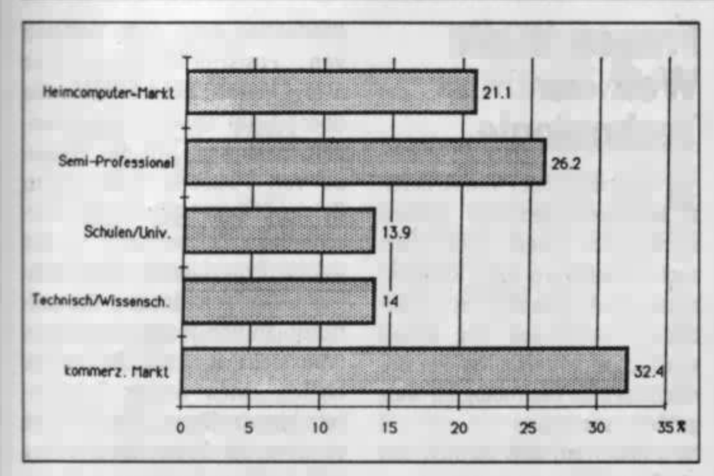
Für die IBM-PC-Familie und alle Kompatiblen hat Paradise die „Hi-Rex Graphics-Card" (PA 31000P) entwickelt. Ihre Kompatibilität umfaßt die IBM-Bildschirm-Adapter CGA und MDA sowie die IBM-fremden Standards Hercules und Plantronics ColorPlus. Die Karte paßt in einen kurzen Steckplatz eines IBM-PC und ist mit einer parallelen Drucker-Schnittstelle ausgestattet. Beim Scrollen des Bildschirms gibt es kein Flimmern mehr. Grundlage der Hi-Res-Karte ist der von Paradise entwickelte Video-Controller-Chip PVC2 und der PPC1 (Parallel Port Chip). Ersterer ist für die Implementierung der Video-Normen zuständig, während letzterer den Aufbau einer Centronics-kompatiblen Druckerschnittstelle mit einem minimalen Bauteileaufwand gestattet. ○



Mehrere Grafik-Standards und eine Drucker-Schnittstelle sind auf dieser kurzen Steckkarte von Paradise untergebracht

Postscript für IBM

Ab sofort steht die Postscript-Software von Adobe Systems Incorporated für alle Modelle der IBM Personal System/2 Serie, die die Micro Channel Architektur aufweisen (Modell 50, 60, 80), zur Verfügung. Mit dem Postscript-Interpreter können auf allen PS/2-Modellen sehr einfach Schriftstücke mit jeder beliebigen Text-/Grafik-Kombination erstellt und gedruckt werden, wobei zwischen einer großen Anzahl von Zeichensätzen gewählt werden kann. Bisher gibt es über 500 Programme, die Postscript verwenden. ○



Gegenwärtige Marktdurchdringung von PCs in der Bundesrepublik Deutschland (Quelle: IDC)

PC-Markt: Kommerzielle Anwendungen an der Spitze

Mit der Marktdurchdringung von PCs in der Bundesrepublik Deutschland befaßt sich eine IDC-Studie. Dabei kommen die Marktforscher zu einem überraschenden Ergebnis: Während der Sättigungsgrad bei kommerziellen Anwendungen derzeit bei 32,4 Prozent liegt, beträgt er im technisch/wissenschaftlichen Bereich lediglich 14 Prozent. Die Vergleichswerte für 1991 lauten: 85 Prozent für kommerzielle und 45 Prozent für technisch/wissenschaftliche Anwendungen. Im Ausbildungssektor soll der Durchdringungsgrad von derzeit 14 Prozent auf über 40 Prozent im Jahr 1991 steigen. Der Heim- und Hobbybereich – derzeit schon zu 30 Prozent gesättigt – kommt nach Ansicht der IDC-Forscher am Ende der betrachteten Periode auf stattliche 57 Prozent. ○

Siemens und Apollo Domain

Siemens und Apollo Domain schlossen ein Rahmenabkommen über 100 Millionen Dollar für die nächsten 12 Monate ab. Apollo liefert an Siemens Computer, auf denen die von Siemens entwickelte Software für das technische Büro lauffähig ist. Die Programme umfassen die gesamten Bereiche des Ingenieurwesens von der Computerunterstützten Software-Entwicklung (CASE) über die technische Dokumentation bis zum mechanischen und elektronischen Design (CAD). Mit dieser Angebotspalette strebt Siemens eine führende Position in diesem wachsenden Markt an. Apollo Domaine ist einer der führenden Hersteller von Unix-Arbeitsplatzcomputern, die hauptsächlich im technisch-wissenschaftlichen Bereich, aber auch zunehmend im kommerziellen Umfeld eingesetzt werden. Mit dem Vertrag wollen Siemens und Apollo ihre Kooperation auf die Bereiche der gemeinsamen Forschung und Entwicklung erweitern. ○



Der Tulip PC compact 2 nimmt wenig Platz in Anspruch

Tulips Mini-PC

„Klein aber oho" ist der richtige Kommentar zum Tulip PC compact 2. Im Gegensatz zu vielen PCs, die wie ein Ei dem anderen gleichen, hat sich hier der Hersteller einiges einfallen lassen:

- Prozessor NEC V20, dessen Taktfrequenz per Software auf 4,77, 7,2 und 10 MHz umschaltbar ist.
- Mausanschluß, Echtzeituhr und Floppy-Controller sind auf der Systemplatine integriert.
- Die Bus-Warte-Zyklen sind vom Anwender programmierbar.
- Die Systemkonfiguration wird wie beim AT in einem CMOS-RAM gespeichert.
- Inklusiv AT-Tastatur MF-II, Schlüsselschalter und Reset-Taste.

Trotz der kleinen Abmessungen (305 × 140 × 375 mm³) hat das System noch fünf freie Steckpätze. In der Geschwindigkeit kommt der Tulip compact 2 fast an die Rechner der 286-Klasse heran. Die Floppy-Version wird mit Microsofts MS-DOS-Manager und die Festplatten-Version mit Windows ausgeliefert. ○

Gateway zwischen dBase und Ingres

Relational Technology (RTI) aus Frankfurt kündigte ein Gateway zwischen dem Datenbanksystem Ingres und der PC-Datenbanksoftware dBase von Ashton-Tate an. Es erlaubt den Zugriff über die Standardabfragesprache SQL transparent auf dBase-DBF-Dateien, als ob es relationale Tabellen wären. In dBase und Ingres geschriebene Programme sollen die gleichen Datenbestände nutzen können. Ohne die bisherigen dBase-Datenbestände aufgeben zu müssen, kann die Ingres-Entwicklungsumgebung der 4. Generation eingesetzt werden. Auch eine vollständige Umstellung von dBase auf Ingres wird damit erleichtert.

Mit dem neuen Gateway wird erstmals eine Brücke zwischen Groß- und Mini-Computern und der PC-Welt geschaffen, die die Investitionen in dBase-Datenbestände schützt.

RTI will noch weitere Gateways zwischen Ingres und anderen Datenverwaltungs-Systemen zur Verfügung stellen, wie z. B. für DB2 und IMS von IBM, RDB von DEC. ○

Frauen in die Welt der Technologie

Arbeitslose oder von Arbeitslosigkeit bedrohte Frauen werden in einem Modellversuch zu Software- bzw. Kommunikations-Assistentinnen ausgebildet. Damit soll der Anteil weiblicher Mitarbeiter im Bereich neuer Technologien vergrößert werden.
Neu dabei ist, daß es sich um ein Fernstudium handelt, das 1 Jahr dauert. Parallel dazu müssen Computerseminare von insgesamt 60 Tagen Dauer besucht werden. 150 Studienplätze stehen pro Lehrgang zur Verfügung. Interessentinnen wenden sich an die Wirtschaftsakademie für Lehrer e.V., Hindenburgring 12 a, 3388 Bad Harzburg, Tel. 0 53 22/7 30. ○

80386-Karte für den PC

Die im Comdex-Bericht (mc 1/88) vorgestellten PC-Karten Intel Inboard 386/PC und das Intel Above Board 2 werden ab sofort von Computer 2000 vertrieben. Mit dem Inboard 386/PC, das unter 2000 DM kostet, kann man einen IBM PC/XT, den Compaq Portable bzw. Portable Plus und den Tandy 1200HD als 386er-Rechner umrüsten. Das Above Board 2 erweitert den Speicherplatz der PS/2-Modelle 50 und 60 um 2 MByte RAM und ist OS/2-kompatibel. ○

Adobe und Linotype kooperieren

Adobe Systems Incorporated, Entwickler der Seitenbeschreibungssprache Postscript, und Linotype, haben ein neues Lizenzabkommen unterzeichnet. Im Rahmen dieses Vertrages wird Linotype ab sofort mit der Entwicklung, der Produktion und dem Vertrieb von Postsript-Versionen der Mergenthaler Type Library, die über 1700 Zeichensätze umfaßt, beginnen. Mit der Freigabe von Postscript-kompatiblen Zeichensätzen stehen 147 verschiedene Zeichensätze in der Adobe Type Library zur Verfügung. Der gemeinsame Produktionsplan von Adobe und Linotype sieht vor, der Bibliothek täglich einen neuen Zeichensatz hinzuzufügen. Zur Zeit gibt es über 20 Postscript-Drucker und Fotosatz-Systeme, und für Adobe ist es von Bedeutung, für diese Produkte eine zunehmende Anzahl von Qualitäts-Zeichensätzen bieten zu können. ○

mc auf der Hobby-tronic

Auf der Hobby-tronic & Computerschau, die vom 3. bis 7. Februar in Dortmund stattfindet, können sich Computeranwender und Hobby-Elektroniker umfassend über das derzeitige Marktgeschehen informieren. Die mc-Redaktion führt eine firmenneutrale Computerberatung durch. Im messenbegleitenden Rahmenprogramm werden Vorträge zu den Themen Satelliten-TV, Desktop Publishing, Datenbanken und mc-modular-AT gehalten. Für Anfänger sind besonders die MS-DOS- und Basic-Einsteiger-Kurse interessant. ○

1,236 GByte-Laufwerk

Nicht nur bei 3½- und 5¼-Zoll-Laufwerken kann man von schnelleren und noch mehr Speicherplatz bietenden Festplatten berichten. Von Control Data kommt das erste 8-Zoll-Festplatten-Laufwerk mit mehr als 1 GByte Kapazität. Das Plattenlaufwerk 1230 aus der Sabre-Familie bietet 1236 MByte freien Speicherraum, hat eine Übertragungsrate von 24,19 MBit in der Sekunde und eine mittlere Positionierungszeit von 16 Millisekunden. ○

Sprachverarbeitung mit LISP und PROLOG auf dem PC

Von Jürgen Handke. 301 Seiten, kart., 64 DM, Vieweg-Verlag, Braunschweig 1987.
ISBN 3-528-04570-1

Allzuoft wendet sich Computer-Literatur an Informatiker oder Naturwissenschaftler. Für Geisteswissenschaftler zum Beispiel wird der Zugang zur modernen Computertechnologie dadurch oft mit unüberbrückbaren Hürden verstellt. Als Linguist mußte dies der Autor Jürgen Handke nach eigenen Angaben leidvoll erfahren, als er sich mit maschineller Verarbeitung von natürlicher Sprache befaßte. Sein Buch ist gut dazu geeignet, ähnliche Annäherungsversuche an die Programmiersprachen Lisp und Prolog optimal zu unterstützen.
Aber auch für durch andere Programmiersprachen Vorgebildete bietet das Buch durch seine Beschränkung auf das für den Lernenden zunächst Wesentliche und Drängendste einen guten Einstieg in die beiden Programmiersprachen. Jedoch sollten sie schon ein gewisses Interesse für natürlichsprachliche Probleme mitbringen, denn die Beispiele entstammen fast ausschließlich diesem Bereich.
Lisp nimmt etwa dreimal soviel Seiten in Anspruch wie Prolog; Parallelen zwische beiden werden jedoch ständig gezogen. So ist auch nach Meinung des Autors weder allgemein noch im Bereich natürlichsprachlicher Phänomeme bereits eine Entscheidung zwischen dem deklarativen Prolog (mit einer Reihe interner Schlußfolgerungsmechanismen) und dem prozeduralen Lisp (ohne interne Schlußfolgerungsmechanismen) gefallen. Der Autor stützt sich auf Common-Lisp, den in Zukunft wahrscheinlichen Lisp-Standard, und den Interface-Prolog-Interpreter Ifprolog. *Br*

Einführung in MS-OS/2

Von Dr. Norbert Meder. Edition Microsoft. 304 Seiten mit zahlreichen Abbildungen, kart., 79 DM, Markt & Technik Verlag, Haar, 1987.
ISBN 3-89090-512-9

Derzeit ist das neue Betriebssystem MS-OS/2 in aller Munde. Für den Anwender oder den an OS/2 Interessierten steht in dem Buch einiges wissenswertes. Einen großen Umfang nimmt die Beschreibung der Befehle des Betriebssystems in Anspruch. In mehrere Teile ist die Beschreibung der Anwender-Befehle gegliedert. Neben den einfachen Kommandos für den täglichen Bedarf, finden sich die Befehle für den fortgeschrittenen Anwender, für den ausgefallenen Bedarf und für den Kenner. Für den Programmierer sind die gegenüber MS-DOS erweiterten Batch-Befehle aufgeführt. Diese Gliederung schützt den Anfänger vor Unheil durch voreiliges Ausprobieren der Befehle für den fortgeschrittenen Anwender. Positiv zu vermerken ist, daß der Autor auch Grundlagenkenntnisse vermittelt. Der Leser wird kurz in die Technik der Speicherverwaltung des Prozessors 80286 eingeführt und auf über zehn Seiten ist die API, die MS-OS/2-Systemschnittstelle für Anwender-Programme, beschrieben. Ein kleines Assembler- und ein C-Programm zeigt auf, wie man unter OS/2 die Systemschnittstelle bedient. Die Beispiele demonstrieren deutlich die Unterschiede gegenüber der MS-DOS-Systemschnittstelle. Zur Darstellung des Multitaskingkonzepts von OS/2 dient die Beschreibung der Bildschirmgruppen, Semaphoren, Threads (Subprozesse), Queues und Pipes. Zum Programmieren reichen diese Informationen nicht aus, einen guten Überblick darüber, was OS/2 kann, bekommt man damit jedoch schon. *St*

dBase einmal ganz verständlich

Janson, Alexander: dBase für Profis. Ausgefeilte Programmier-Tricks für Insider mit dBase II, III und III plus, 272 Seiten, geb., 68 DM.
ISBN 3-7723-8871-X
Janson, Alexander: dBase II, III und III plus kompakt, Kurzhandbuch der beiden Datenbanksysteme, kart. 144 Seiten, Franzis-Verlag, 16,80 DM.
ISBN 3-7723-2032-5

Wer in dBase II, III oder III plus programmieren möchte, kann jetzt auf zwei Bücher von Alexander Janson zurückgreifen. In dem Buch „dBase für Profis" beschreibt der Autor die effiziente Arbeitsweise mit dieser Programmiersprache. Er vermittelt Grundwissen zu fortgeschrittenen Techniken des Programmierens und liefert gleich pfiffige Tips und Tricks zur Lösung bestimmter Aufgaben mit.
Alle Schritte zwischen Pflichtenheft und Software-Pflege sind durch zahlreiche Beispiele belegt. Die Programmierung von problemunabhängigen Routinen und einfache Lösungen für eine Geschwindigkeitssteigerung sind weitere interessante Themen. Auch der Einsatz eines Compilers wird erläutert.
Eine gute Ergänzung zu diesem Buch ist das Kompendium zu den dBase-Versionen. In alphabetischer Reihenfolge sind alle Befehle aufgeführt und kurz beschrieben.
Clipper-Kommandos und das Koppeln von dBase mit WordStar, MailMerge und mit Basic finden als wichtige Bestandteile der praktischen Arbeit mit den Datensätzen in beiden Büchern gebührend Beachtung.
Bemerkenswert ist die Selbstverständlichkeit, mit der Alexander Janson komplizierte Vorgänge logisch, ohne viel Fachjargon und mit einfachen Worten verständlich schildert. Zwei Bücher, die ich empfehle.

Lore Schönrock

Hard- und Software-Tuning für IBM PCs und Kompatible

Herausgegeben von U. Moser, Ringbuchordner DIN A4 mit etwa 400 Seiten, Interest-Verlag, Kissing, 92 DM.
Bestell-Nr. 2600.

Trotz der Vielfalt von Büchern über MS/PC-DOS fehlte bisher für den Einsteiger ein Lern- und Nachschlagewerk, das Hard- und Software zugleich behandelt. Diese Lücke versucht das Arbeitsbuch von U. Moser zu schließen. Alle 2 bis 3 Monate wird das Werk durch Nachlieferungen ergänzt und aktualisiert. Ausgehend vom IBM-PC-Standard wird der Leser vom Aufbau eines PC über die Beschreibung der CPU 8088/86 und des Betriebssystems zur Anwendung verschiedener Programmiersprachen geführt. Der mit 32 Seiten Quelltext umfangreichste Teil beschreibt die Anwendung eines in Turbo-Pascal geschriebenen Assemblers, der auf Diskette erhältlich ist. Prinzipien der Softwareentwicklung und ein Kurs in Sachen Datenverwaltung vermitteln viel Grundwissen. Dabei wird, obwohl an anderer Stelle davor gewarnt wird, mit Basic gearbeitet. Beispiele zu Grafik und Mathematik runden den Softwareteil ab. Auch Tips zur Erweiterung des Rechners und den unterschiedlichen DOS-Versionen sind enthalten. Nach der Beschreibung der MS-DOS-Befehle wird das Arbeiten mit DEBUG und mit Batch-Dateien erläutert. Dabei werden Hinweise zu DOS-Aufrufen unter Turbo-Pascal gegeben. Einen weiteren Schwerpunkt bilden die PC-Erweiterungen: Einbau einer Festplatte, mathematische Coprozessoren, der Selbstbau einer parallelen I/O-Karte und eines Bild-Digitalisierers mit Software sowie Grundlagen über Netzwerke und DFÜ.

Dr. Hans Hehl

Dieter Strauß

Ein erster Blick auf MS-OS/2

Multitasking für Rechner der AT- und 386er-Klasse

Für PCs wurden Multitasking-Betriebssysteme bereits von verschiedenen Software-Häusern entwickelt. Mit MS-OS/2 kommt ein Multitasking-Betriebssystem für Rechner der AT- und 386er-Klasse auf den Markt, das gute Chancen hat, eine ähnlich weite Verbreitung wie MS-DOS zu finden.

Bereits schon 1982 kam mit QNX ein Multitasking-Betriebssystem für PCs auf den Markt. Sogar von Microsoft gibt es bereits seit geraumer Zeit das Multiuser-/Multitasking-Betriebssystem Xenix, das ebenfalls wie OS/2 mindestens einen Rechner der AT-Klasse voraussetzt.

QNX und Xenix orientieren sich an Unix, das zwar ein sehr mächtiges, aber auch gleichzeitig kein einfach zu bedienendes Betriebssystem ist.

Wie sich Betriebssysteme, die für die breite Masse der Anwender geeignet sind, präsentieren sollten, hatte Xerox schon in den siebziger Jahren gezeigt:

Eine grafische Bedieneroberfläche, bei der man Menüpunkte einfach mit der Maus anklickt.

Apple hatte mit dem Macintosh diese Idee zu einer Zeit verwirklicht, als PC-Anwender noch nicht einmal die ausgefeilte zeichenorientierte Benutzeroberfläche kannten, die z. B. Borland bei Turbo-Prolog verwendet.

Betriebssysteme mit grafischer Bedieneroberfläche sind für Anwendungen geeignet, bei denen der Mensch-Maschinen-Dialog möglichst einfach gestaltet sein soll, damit auch Anwender, die einen Computer nur gelegentlich benutzen, ohne „Informatik-Studium" klarkommen.

Solche Betriebssysteme haben von den System-Resourcen her gesehen zwei Nachteile:

- Sie sind sehr hungrig auf die CPU-Leistung. Ein großer Teil davon wird vom Betriebssystem selbst beansprucht.
- Jede Menge Speicherplatz wird benötigt. Unter einem MByte Speicher ist fast nichts drin.

Aus diesen Gründen dürften sich Microsoft und IBM entschieden haben, für das neue Betriebssystem mindestens einen Rechner der AT-Klasse vorauszusetzen. Bei der Codierung von OS/2 konnten somit die neuen Befehle des 80286-Prozessors und dessen geschützte Betriebsart (protected mode) verwendet werden.

Zwei Forderungen schließen sich bei einem Betriebssystem mit grafischer Bedieneroberfläche dagegen fast aus:

Multitasking- und Multiuser-Fähigkeit.

Selbst die CPU 80286 geht bei einem Multiuser-/Multitasking-Betriebssystem mit grafischer Benutzeroberfläche in die Knie.

Weiterhin sind die bei einem Multiuser-Betriebssystem mit grafischer Benutzeroberfläche notwendigen Grafikterminals nicht gerade billig. Die Schöpfer von OS/2 haben das erkannt und daher auf die Mehrplatz-Fähigkeit verzichtet.

Damit hält sich Microsoft an das Postulat des amerikanischen Computerexperten Jerry Pournelle [1], der für einen Anwender mindestens (!) eine CPU fordert.

Nicht verwirklicht ist in OS/2 das Multiprocessing, das parallel ablaufende Tasks auf mehrere Prozessoren verteilt. Für ein Massenbetriebssystem wie MS-OS/2, das auf preiswerten Rechnern laufen soll, wäre das zumindest derzeit nicht realisierbar.

Die Entwicklung von MS-OS/2 ist noch nicht abgeschlossen

Die Entwicklung eines neuen Betriebssystems, das zudem eine Kompatibilitätsbox für MS-DOS-Programme haben soll, dauert eine Weile. Da die Forderungen der Anwender nach einem einfach zu bedienenden Betriebssystem für den PC immer lauter werden und schließlich der Wettbewerb nicht schläft, haben sich Microsoft und IBM entschlossen, möglichst schnell mit dem Betriebssystem-Kern auf den Markt zu kommen. Da es einige Verwirrungen um die verschiedenen OS/2-Versionen gibt, fassen wir einige bereits von uns veröffentlichte Informationen nochmals zusammen.

Die MS-OS/2-Versionen

MS-OS/2 Version 1.0: Diese Version hat die gleiche Kommandozeilen-orientierte Benutzeroberfläche wie MS-DOS. Auch wird eine Kompatibilitätsbox bereitgestellt, in der die meisten MS-DOS-Programme laufen dürften. Sogar das direkte Schreiben in den Bildschirmspeicher ist erlaubt. Die Kompatibilitätsbox setzt sämtliche MS-DOS-Funktionsaufrufe während des Programmlaufs in OS/2-Funktionsaufrufe um. Kompatibilität gibt es natürlich nicht zum Nulltarif: Die Umsetzung kostet – wie könnte es auch anders sein – CPU-Zeit. MS-DOS-Anwendungen laufen unter OS/2 daher etwas langsamer ab.

Neben den bekannten MS-DOS-Befehlen gibt es einige neue Befehle:

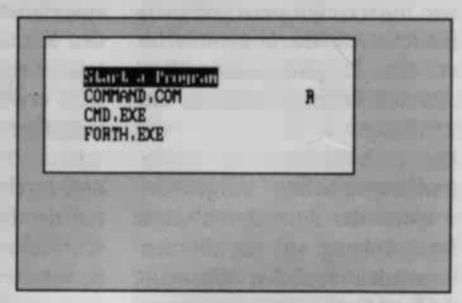


Bild 1. So präsentiert sich der Session-Manager auf dem Bildschirm

- ANSI: Treiber für ANSI-Sequenzen im Protected Mode installieren. Entspricht ANSI.SYS unter MS-DOS.
- CMD: Zweiten Kommandoprozessor im Protected Mode starten.
- DETACH: Prozeß soll im Hintergrund laufen.
- DPATH: Legt einen Suchpfad für Dateien an, die keinen ausführbaren Code enthalten und auch keine Batch-Datei sind. Entspricht „append" im Real Mode
- HELPMSG: Erläutert Fehlermeldungen des Betriebssystems.
- PATCH: An Programmen und Dateien werden Änderungen durchgeführt. Offset und Daten muß man im Hex-Code eingeben.

Auch einige Batch-Befehle sind neu:

- CALL: Diesen Befehl gibt es bereits unter PC-DOS 3.3. Damit kann eine Batch-Datei eine andere aufrufen. Sind alle Befehle der aufgerufenen Batch-Datei ausgeführt, kehrt das System wieder in die aufrufende Batch-Datei zurück und führt deren restlichen Befehle aus.

- EXTPROC: Aktiviert einen externen Batch-Kommandoprozessor. Er ist nur so lange aktiv, bis alle Befehle der Batch-Datei ausgeführt sind.
- SETLOCAL: Zur Bearbeitung von Batch-Befehlen kann man den Umgebungsvariablen neue Werte zuweisen und ein anderes aktuelles Directory und Laufwerk definieren.
- ENDLOCAL: Der Zustand vor dem ENDLOCAL-Befehl wird wiederhergestellt.

Batch-Dateien, die in der geschützten Betriebsart (Protected Mode) arbeiten, sind mit der Dateinamenerweiterung CMD gekennzeichnet. Im Real Mode betriebene Batch-Dateien sind an der Endung BAT erkennbar. Mit der Tastenkombination Ctrl-Esc schaltet man zwischen der MS-DOS-Kompatibilitätsbox und dem sogenannten Session-Manager um.

MS-OS/2 Version 1.1: Bei dieser OS/2-Version ist der Windows Presentation Manager – die grafische Bedieneroberfläche – dem Betriebssystemkern hinzugefügt. Allerdings wird der Windows Presentation Manager erst im Herbst auf den Markt kommen. Im Aussehen und in der Bedienung wird er Windows 2.0 gleichen.

MS-OS/2 Erweiterte Version: Das neue Betriebssystem wurde von Microsoft und IBM gemeinsam entwickelt. Die erweiterte Version (Extended Version) gibt es nur für IBM-Rechner. Damit die IBM ein zusätzliches Verkaufsargument für ihre Systeme hat, hat sie ein Datenbank- und ein Kommunikationsprogramm in das Betriebssystem integriert. Da es jedoch einzeln zu kaufende Datenbank- und Kommunikationsprogramme unter OS/2 gibt, sind die Anwender von Standard-OS/2 nicht sonderlich benachteiligt.

Familien-Applikationen: Familien-Applikationen sind Programme, die eine Untermenge der Anwendungsprogramm-Schnittstelle (API: Application Programm Interface) verwenden, d. h. sie rufen nur die OS/2-Funktionen auf, für die es äquivalente MS-DOS-Funktionsaufrufe gibt.

Kernel-Applikationen: Vollen Zugriff auf alle OS/2-Funktionen haben die Kernel-Applikationen, die daher auch nur unter OS/2 laufen können. Damit kann der Programmierer das ganze Arsenal der Multitasking-Eigenschaften des OS/2 ausnutzen.

Der Session-Manager

Solange OS/2 noch keinen Windows Presentation Manager hat, muß man sich mit der zeichenorientierten Benutzeroberfläche begnügen. Mit dem Session-Manager startet man Programme und schaltet zwischen den verschiedenen Bildschirmgruppen um. Er wird mit der Tastenkombination Ctrl-Esc aufgerufen. Auf dem Bildschirm sieht man darauf eine Liste der bereits gestarteten Programme (*Bild 1*). Will man in die Bildschirmgruppe eines Programms gelangen, fährt man mit den Pfeiltasten den jeweiligen Pogrammnamen an und betätigt die Eingabetaste. Ebenfalls mit der Tastenkombination Ctrl-Esc kommt man aus einem Programm in den Session-Manager zurück. Um zwischen den einzelnen Programmen umzuschalten, kann man auch die Tastenkombination Alt-Esc verwenden. Bei mehrfacher Betätigung dieser Tasten gelangt man der Reihe nach durch alle Bildschirmgruppen. Die Reihenfolge entspricht der im Menü des Session-Managers aufgeführten Programme.

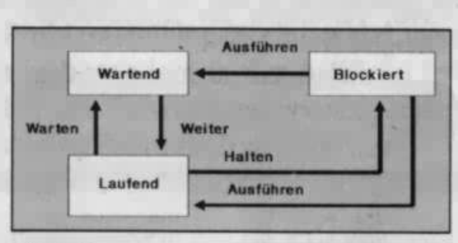


Bild 2. Übergangsgraph eines Threads

Betriebssystem-Technik

OS/2 ist ein Multitasking-Betriebssystem, das auf dem prioritätengesteuerten Zeitscheiben-Konzept beruht. Jedem Prozeß wird eine kleine Zeitspanne zugeteilt, während der er die CPU benutzen darf. Zur Zeitsteuerung verwendet das System einen Interrupt des Zeitgeber-Bausteins 8253/8254. Nach Ablauf der Zeitscheibe löst der Zeitgeber einen Interrupt aus. Mit diesem Interrupt wird der sogenannte Dispatcher angestoßen. Er sichert die aktuellen Registerinhalte, damit er später den Prozeß an der unterbrochenen Stelle wieder fortsetzen kann. Eine weitere Betriebssystemfunktion ist der Scheduler (engl.: schedule = Fahrplan), der bestimmt, welcher wartende Prozeß aufgrund seiner Priorität an der Reihe ist und den Dispatcher anweist, diesen zur Ausführung zu bringen. Die Umschaltung zwischen zwei Prozessen bezeichnet man als Kontextwechsel. Von all dem merkt der Anwender nichts. Beim Kontextwechsel kann die CPU 80286 eine ihrer Stärken ausspielen. Ihre Schöpfer haben nämlich dafür gesorgt, daß der Kontextwechsel stark von der Prozessor-Hardware unterstützt wird.

Neben Prozessen existieren unter OS/2 noch Bildschirmgruppen und Threads.

Unter einem Prozeß versteht man in diesem Zusammenhang eine durch das Anwenderprogramm spezifizierte Folge von Aktionen, deren erste begonnen, deren letzte aber noch nicht abgeschlossen ist [1].

Bildschirmgruppen und Threads

Einer Bildschirmgruppe sind jeweils ein oder mehrere Prozesse zugeordnet. Zu den nicht aktiven Bildschirmgruppen gehören virtuelle Bildschirme und Tastaturen. Mit diesem Konzept wird vermieden, daß die unzähligen Prozesse den Bildschirm mit ihren Nachrichten so vollkritzeln, daß der Anwender überhaupt nicht mehr weiß, was gespielt wird.

Unter OS/2 kann ein Prozeß noch Subprozesse laufen lassen. Jedem Subprozess oder Thread sind ein eigener Stack, die jeweiligen Registerinhalte und ein Prozeßzustand zugeordnet. OS/2-Threads kennen die Zustände blockiert, wartend und laufend (*Bild 2*). Ein blockierter Thread wartet auf ein bestimmtes Ereignis, z. B. darauf, daß er auf einen von mehreren Threads gemeinsam verwendeten Datenbereich zugreifen darf. Ein wartender Thread ist bereit zur Ausführung. Erst wenn ihm der Scheduler eine Zeitscheibe zuteilt, beginnt er mit der Ausführung seines Codes.

Es ist wesentlich, daß alle Threads eines Prozesses auf das gleiche Daten-Segment zugreifen. Der Programmierer muß daher den Zugriff dieser „konkurrierenden" Threads auf gemeinsame Daten synchronisieren. Ein geeignetes Mittel sind dazu Semaphoren. Eine Semaphore kann man, salopp formuliert, als eine Besetzt-Anzeige bezeichnen. Greift gerade ein Thread auf den gemeinsam genutzten Datenbereich zu, zeigt die Semaphore den Zustand „besetzt" an. Ein quasi parallel laufender Thread greift erst auf diesen Datenbereich zu, wenn die Semaphore den Zustand „frei" anzeigt.

Ein Programm, das unter OS/2 Daten von der Festplatte liest, sie verarbeitet und schließlich die Ergebnisse auf dem Bildschirm ausgibt, könnte vom Programmierer in je einen Thread zur Bearbeitung des Hauptprogramms, zum Lesen der Daten von der Festplatte, zum Verarbeiten der Daten und zur Ausgabe der Ergebnisse auf dem Bildschirm unterteilt werden. Beim Programmaufruf erzeugt das Betriebssystem den Thread des Hauptprogramms, der daraufhin die entsprechenden Semaphorvariablen initialisiert und die restlichen Threads generiert.

Funktionsaufrufe ohne Interrupts

Ein Betriebssystem stellt eine Sammlung von Steuerungsprogrammen und Hilfsroutinen dar, welche die Anwendung eines

Rechners und der daran angeschlossenen Geräte für den Menschen vereinfachen. Eine der Hauptaufgaben eines Betriebssystem ist es, ein E/A-System für die Anwendungsprogramme zur Verfügung zu stellen. Unter MS-DOS wird dies in erster Linie vom Interrupt 21H durchgeführt.

Viele der 256 Interrupts der 8086-Prozessor-Familie werden von MS-DOS verwendet und auch den Anwendungsprogrammen als Systemschnittstelle angeboten.

Unter MS-DOS macht fast jedes Programm von mehreren Interrupts Gebrauch und viele Programme ändern Interrupt-Vektoren. So lange jedes Programm nach Beendigung wieder den Zustand herstellt, der vor seinem Aufruf bestand, ist gegen diese Methode in einem Single-Task-Betriebssystem nichts zu sagen. Kritisch wird die Sache erst, wenn man mehrere speicherresidente Programme verwendet. Da kein Programm sich um das andere kümmert, kommt es häufig vor, daß zwei speicherresidente Programme den gleichen Interrupt-Vektor verwenden. Die Folge sind oft rätselhafte Systemabstürze.

In einem Multitasking-Betriebssystem wäre eine Systemschnittstelle, die auf Interrupts beruht, sehr instabil. Die Schöpfer von OS/2 haben sich daher entschlossen, einen völlig anderen Weg zu gehen. Wir nehmen daher Abschied von folgenden für MS-DOS typischen Befehlssequenzen:

```
MOV AX, 4CH
INT 21H
```

Stattdessen steht unter OS/2 ein Mechanismus zur Verfügung, der dem Aufruf von Unterprogrammen unter Hochsprachen wie Pascal und C ähnlich ist. In einem Assembler-Programm verwendet man einfach den CALL-Befehl. Der oben gezeigte Code sieht unter OS/2 so aus:

```
CALL DOSEXIT
```

Der eigentliche Code zu diesem Betriebsystemaufruf steht im OS/2-Code-Segment, d. h. der CALL-Befehl ist genaugenommen ein FAR CALL (far = entfernt).

Bei den meisten Betriebssystemaufrufen muß man Parameter übergeben. Unter MS-DOS werden zur Parameterübergabe die verschiedenen Prozessor-Register verwendet während unter OS/2 die Parameter – wie z. B. bei Pascal und C üblich – vor dem Funktionsaufruf einfach auf den Stack gelegt werden.

Ein OS/2-Funktionsaufruf zerstört nur den Inhalt des AX-Registers und die Flags. Bei der Rückkehr ins Anwendungsprogramm räumt das Betriebssystem die Parameter vom Stack. Zur Kontrolle, ob der Funktionsaufruf erfolgreich durchgeführt wurde, gibt die jeweilige Funktion im AX-Register einen Fehlercode zurück. Ist dieser Null, wurde die Funktion erfolgreich ausgeführt. Microsoft hat sich für die Benennung der Funktionsaufrufe eine Konvention ausgedacht. Alle OS/2-Aufrufe, die zum Betriebssystemkern gehören, kann man den folgenden Gruppen zuordnen:

- DOS: Dateiverwaltung, Speicherzuteilung, Multitasking
- KBD: Tastatur-Funktionen
- MOU: Funktionen zur Unterstützung der Maus
- VIO: Funktionen zur Bildschirm-Ein-/Ausgabe (Video I/O)

```
Dynamic link libraries:

     MSG        DLL      KBDCALLS DLL
     NLS        DLL      MONCALLS DLL
     ANSICALL   DLL      MOUCALLS DLL
     BKSCALLS   DLL      QUECALLS DLL
     BMSCALLS   DLL      SESMGR   DLL
     BVSCALLS   DLL      VIOCALLS DLL
     DOSCALL1   DLL
```

Bild 3. Das Betriebssystem verwendet selbst dynamische Link-Bibliotheken

Hochsprachenähnliche OS/2-Funktionsaufrufe

Ein Anwenderprogramm, das Betriebssystemfunktionen aufruft, kennt nicht ihre Adressen. Wenn Programme assembliert bzw. compiliert und gelinkt sind, stehen immer noch die Funktionsaufrufe ohne Adreßbezug da.

Erst wenn OS/2 ein Anwenderprogramm in den Speicher lädt, setzt es die tatsächlichen Adressen ein. Dieser Vorgang wird als dynamisches Linken bezeichnet. Die OS/2-Funktionsaufrufe sind in dynamischen Link-Bibliotheken (dynamic link libraries) gespeichert. Dateien mit dynamischen Link-Bibliotheken erkennt man an der Dateinamenerweiterung DLL.

Bild 3 stellt ein Directory mit mehreren DLL-Dateien dar. Da jede dynamische Link-Bibliothek eine Erweiterung des Betriebssystems ist, können Software-Häuser leicht Betriebssystem-Erweiterungen entwickeln, die von mehreren Anwenderprogrammen genutzt werden.

Eine Inkompatibilität muß der Anwender in Kauf nehmen: Das Dateiformat einer EXE-Datei ist verschieden gegenüber einer MS-DOS-EXE-Datei.

Viele Leser werden jetzt bereits einen Nachteil des dynamischen Linkens bemerkt haben: die Ladezeit eines Programms erhöht sich. Da bei einem Multitasking-Betriebssystem mehrere Programme quasi simultan laufen können, wiegt dieser Nachteil nicht so schwer. Nach dem Einschalten des Systems kann man nacheinander die benötigten Programme laden. Wenn es sich nicht gerade um rechenintensive Programme handelt, bleibt dennoch die Antwortzeit des Systems akzeptabel. Wunder kann ein Multitasking-Betreibssystem jedoch nicht vollbringen; schließlich steht für alle Prozesse nur eine CPU zur Verfügung.

Aus für COM-Dateien

COM-Dateien gibt unter OS/2 nur noch in der Kompatibilitätsbox. Die Größe eines COM-Programms ist auf 64 KByte begrenzt, was allein wegen der heutigen Ansprüche an die Qualität der Benutzerführung bei kommerziellen Programmen nicht mehr ausreicht. Es gibt aber noch einen weiteren Grund: bei COM-Dateien sind der Programm-Code und die Daten im gleichen Segment. Unter OS/2 wird jedoch der Prozessor im geschützten Modus (Protected Mode) betrieben, der voraussetzt, daß Programm-Code und Daten voneinander getrennt sind.

Intel hat die Prozessoren 80286 und 80386 im geschützten Modus mit vier Schutzebenen versehen. OS/2 verwendet drei Schutzebenen:

- Ebene 0 für das System
- Ebene 2 für spezielle E/A-Routinen
- Ebene 3 für den Anwender

Die Ebene 1, die Intel für System-Dienste vorsah, wird nicht verwendet. Microsoft hat damit ein Dreistufen-Schutzkonzept gewählt, das jedenfalls in Betriebssystemen der 68000-Familie nicht möglich ist.

Einem Segment ist eins aus drei möglichen Attribute zugeordnet. Die Attribute lauten:

- executable: Es handelt sich um ein Code-Segment
- read-only: Ein Daten-Segment, das nur gelesen werden darf
- read-write: Ein Daten-Segment, das gelesen und beschrieben werden kann

Mit diesem Schutzkonzept dürfte in Zukunft der Reset-Taster an PCs nicht mehr so häufig wie bisher betätigt werden. Nur noch außer Kontrolle geratene MS-DOS-Programme in der Kompatibilitätsbox können das System relativ leicht abstürzen lassen.

Dieser Beitrag konnte die Themen, die unter OS/2 aktuell werden, nur schlaglichtartig beleuchten. mc wird in lockerer Folge über das neue Betriebssystem berichten.

Literatur

[1] *Pournelle, Jerry:* On the road to Karlsruhe. Byte, November 1987

[2] *Weck, Günther:* Prinzipien und Realisierung von Betriebssystemen. B. G. Teubner Stuttgart

Klaus Manhart

Prolog und Logik

Der Beweismechanismus von Prolog und seine logischen Grundlagen (Teil 1)

Prolog ist die derzeit höchste Programmiersprache, die ganz allgemein für die Bearbeitung nichtnumerischer Problemstellungen eingesetzt werden kann und sich insbesondere in der Artificial Intelligence – etwa bei der Entwicklung von Expertensystemen – wachsender Beliebtheit erfreut.

Seinen Aufstieg verdankt Prolog unter anderem der Entscheidung, daß bei dem ehrgeizigen japanischen Forschungsprojekt, die nächste „intelligente" Computergeneration zu entwickeln, Prolog als Basissprache eingesetzt wird. Für die Zukunft kann die Bedeutung von Prolog also kaum überschätzt werden. Aber bereits heute werden zahlreiche Expertensysteme und Programme, die natürliche Sprache verarbeiten, in Prolog geschrieben und nicht mehr mit dem „traditionellen" Artificial-Intelligence-Instrument Lisp.

Prolog beruht auf der klassischen Prädikatenlogik und steht für Programming in Logic. In der Tat wird eine Variante dieser Logik direkt als Programmiersprache verwendet, so daß sich grundlegende Unterschiede ergeben zu prozeduralen, algorithmischen Sprachen wie Fortran oder C, aber auch zu funktionalen wie Lisp. Prolog ist, zumindest in seiner reinen Form, völlig „anti-algorithmisch". Damit ist gemeint, daß der Programmierer dem System zur Lösung eines Problems keinerlei Anweisungen im herkömmlichen Sinn mehr vorzuschreiben braucht, sondern die Struktur des Problems lediglich zu beschreiben hat. Prolog sucht sich seine Lösung „eigenständig" mit einem Deduktionsmechanismus, dessen Grundlagen und Arbeitsweisen dieser Artikel beschreibt.

Ein kleines Programm-Beispiel

Die Interaktion mit einem Prolog-Interpreter oder Compiler besteht im Grunde in nichts anderem als der Eingabe von Wissen und dessen Nutzbarmachung in Form von Fragen. Das folgende Beispiel zeigt ein – allerdings sehr kleines und primitives – Prolog-Programm (die Numerierung gehört nicht zum Programm):

(1) kind_von(wolfgang_amadeus_mozart, leopold_mozart).
(2) maennlich(leopold_mozart).
(3) vater_von(X,Y) :- maennlich(X), kind_von(Y,X).

Das Programm besteht aus drei Formeln, die auch Clausen genannt werden und immer mit einem Punkt abzuschließen sind. Die Clausen (1) und (2) sind Fakten, mit denen dem System die für ein Problemgebiet relevanten Sachverhalte mitgeteilt werden und die vom Benutzer als wahr anerkannt werden. Im vorliegenden Fall sollen die beiden Fakten besagen, daß Wolfgang Amadeus Mozart das Kind von Leopold Mozart ist und daß Leopold Mozart männlich ist.

Die Underliner „_" haben keine besondere Bedeutung sondern dienen lediglich der besseren Lesbarkeit. Clause (3) ist eine Regel. Regeln drücken allgemeine, für das Problemgebiet wichtige Zusammenhänge aus, im obigen Beispiel:

Für alle X und Y gilt: Wenn X männlich ist und Y das Kind von X ist, dann ist X der Vater von Y.

Man kann erkennen, daß der Dann-Teil einer Regel – auch Kopf genannt – in Prolog zuerst hingeschrieben werden muß und von den Bedingungen – dem Rumpf der Regel – durch die Zeichenfolge :- getrennt wird. Der Kopf darf nur aus einem Element bestehen, während der Rumpf mehrere z. B. durch „und" verknüpfte Bedingungen beinhalten darf; durch „und" verknüpfte Bedingungen sind mit Kommas zu trennen. Schließlich kommen in Regeln meist noch Variable vor. Variable sind Zeichenketten, die mit Großbuchstaben beginnen und Platzhalter für konkrete Objektnamen sind. Der Grund für die Notationsweise von Regeln wird weiter unten deutlich.

Die Fakten bilden zusammen mit den Regeln die Wissensbasis eines Programms, an die Fragen gestellt werden können. Fragen beginnen in Prolog mit ?-. Die Frage

?- maennlich(leopold_mozart).

beantwortet das System mit

Yes

Die Frage

?- maennlich(wolfgang_amadeus_mozart).

wird mit

No

beantwortet.

Prolog prüft dabei, ob der durch eine Frage ausgedrückte Sachverhalt logisch mit der Wissensbasis vereinbar ist. Genauer gesagt, versucht es zu zeigen, daß ein Theorem (die Frage) logisch aus den Axiomen (Fakten und Regeln) folgt. Eine Frage wird deshalb auch als Ziel bezeichnet, das zu beweisen ist. Beispielsweise ist das zweite Beweisziel

maennlich(wolfgang_amadeus_mozart)

in der realen Welt zwar richtig, kann aber aus den vorliegenden Fakten und Regeln nicht bewiesen werden. Aufgrund seiner Eigenschaft, Theoreme aus Axiomen abzuleiten, kann man Prolog auch als automatischen Theorembeweiser bezeichnen.

Interessantere Ableitungen ergeben sich, wenn das System nicht nur mit Ja oder Nein reagiert, sondern Antworten generiert. Mit

?- vater_von(X,wolfgang_amadeus_mozart).

kann etwa gefragt werden, wer der Vater von W. A. Mozart ist. Wie unschwer zu erraten ist, antwortet das System mit

X = leopold_mozart

Die Frage

?- vater_von(X,Y).

wird mit

X= leopold_mozart
Y= wolfgang_amadeus_mozart

beantwortet. Bei den letzten beiden Fragen ist zu erkennen, daß Prolog richtige Antworten gibt, die nicht explizit in der Wissensbasis stehen.

Die Sprache der Prädikatenlogik

Die Kenntnis der logischen Grundlagen und der Lösungswege ist eine unabdingbare Voraussetzung für das ernsthafte Arbeiten mit Prolog. Die nicht immer einfachen theoretischen Ausführungen werden dabei durch Beispiele illustriert.

Die Prädikatenlogik erster Stufe, auch Quantorenlogik genannt, geht in ihren Wurzeln bis auf Aristoteles zurück und wurde in ihrer heutigen Form in der Hauptsache von Gottlob Frege und Bertrand Russel begründet. Die Beschäftigung mit Logik war bis zur Einvernahme bestimmter Logik-

konzepte durch die Artificial Intelligence ein eher esoterisches Gebiet der Mathematik und Philosophie ohne konkrete Anwendungsmöglichkeiten, abgesehen vielleicht von Informatikgrundlagenforschung. Mit dem Siegeszug der Artificial Intelligence und der Entwicklung von Logikprogrammiersprachen änderte sich dies jedoch grundlegend.

Bei der Umsetzung eines Problems in die präzise logische Sprache hat man meist eine umgangssprachliche Formulierung im Kopf. Tatsächlich besteht zwischen Logik und natürlicher Sprache eine enge Beziehung, es existieren aber auch viele Unterschiede und Schwierigkeiten, wenn man versucht, natürliche Sätze in die asketische logische Sprache zu transformieren.

Das wichtigste Grundprinzip der klassischen Logik ist das Bivalenzprinzip. Es besagt ganz einfach: Jeder Satz ist entweder wahr oder falsch. Theoretisch teilt die Logik dieses Prinzip mit (Aussage-)Sätzen der natürlichen Sprache, ist dort aber nur mit erheblichen Einschränkungen und Schwierigkeiten anwendbar. Fast alle natürlichsprachlichen Sätze sind nämlich mehr oder weniger vage. Den Satz „Europa ist groß" wird man beispielsweise kaum als wahr oder falsch schlechthin betrachten. In einem gegebenen Kontext, bei gegebenen Maßstäben, Definitionen, Erwartungen, Hörern etc. können Sätze dieser Art aber durchaus wahr oder falsch sein.

Eine der wichtigsten Anwendungsvoraussetzungen der klassischen Logik ist deshalb:
Jedes Prädikat ist hinreichend scharf.

Ein Prädikat ist die Eigenschaft, die einem Objekt zugeschrieben ist (im Beispiel: „..ist groß") bzw. die Relation, die zwischen Objekten besteht.
Die klassische Logik fordert somit, daß man immer entscheiden kann, ob ein Prädikat („...ist groß") auf ein bestimmtes Objekt („Europa") zutrifft oder nicht. Erwähnenswert hierzu ist noch, daß in neuerer Zeit Logiken entwickelt wurden, die sowohl das Bivalenzprinzip fallenlassen (drei- und mehrwertige Logiken) als auch die Forderung der Schärfe der Prädikate (Fuzzy Logiken). Für Prolog hat dies allerdings keine Bedeutung.
Eine weitere Gemeinsamkeit von Logik und natürlicher Sprache ist: beide haben ein Vokabular, eine Syntax und eine Semantik. Anders als in der natürlichen Sprache sind in der Logik aber Zeichenvorrat, Syntax und Semantik präzise definiert.

Die Prädikatenlogik ist induktiv aufgebaut

Ausgehend von den kleinsten Bestandteilen werden immer komplexere Ausdrücke definiert. Der Zeichenvorrat der Prädikatenlogik besteht dabei aus folgenden Symbolen, wobei in Hinblick auf Prolog die Notation von der Konvention abweicht:

- ein Prädikat ist eine (endliche, nichtleere) Folge von Zeichen des lateinischen Alphabets ohne Leerzeichen und mit Unterstrich _, beginnend mit einem Kleinbuchstaben (Beispiele: a, b, pred, klein_buch_stabe). Man unterscheidet 0-,1-,2-,...,n-stellige Prädikate und gibt damit die Zahl der Argumente an, die ein Prädikat verlangt. In der natürlichen Sprache ist beispielsweise das Prädikat
  ... liebt ...
  2stellig, da es zwei Argumente erfordert, das Prädikat
  ... groß
  ist einstellig.
- eine Variable ist definiert wie ein Prädikat, außer daß der erste Buchstabe der Zeichenfolge mit Großbuchstaben beginnt (Beispiele: X, Y, Variable, Pred).
- Konstanten und Funktoren sind definiert wie Prädikate.
- logische Konstanten: ¬ (nicht), & (und), ∨ (oder), => (wenn-dann), <=> (genau dann, wenn)
  - Quantorzeichen: ∀, ∃
  - Sonderzeichen: Klammern, Doppelpunkt, Komma.

Aus diesen Grundbausteinen werden nun in der Syntax Terme, Atomformeln und komplexe Formeln (molekulare Formeln) gebaut.
Terme sind induktiv definiert:
(T1) Konstanten und Variable sind Terme.
(T2) Ist $\underline{f}$ ein Funktor und sind $\underline{t}_1,...,\underline{t}_n$ Terme, so ist $\underline{f}(\underline{t}_1,...,\underline{t}_n)$ ein Term. Dabei sind $\underline{t}_1,...,\underline{t}_n$ die Argumente von $\underline{f}$.
Die unterstrichenen Symbole gehören nicht zur zu definierenden Sprache, sondern dienen hier und im folgenden immer lediglich als Platzhalter für die Ausdrücke der Sprache. Terme kann man in der Umgangssprache mit Objektnamen vergleichen. Terme der letzten Art entsprechen dabei Konstrukten wie „der Verfasser von Ulysses", „die Summe von zwei und drei", in Logikschreibweise sieht das so aus:
verfasser_von(ulysses).
summe_von(zwei,drei).
Terme bezeichnen immer Objekte, die nicht wahr oder falsch sein können; das sollte man nie vergessen.
Der nächste Schritt besteht nun darin, daß man aus Termen und Prädikaten Atomformeln bildet:

(A) Ist $\underline{p}$ ein n-stelliges Prädikat und sind $\underline{t}_1,..,\underline{t}_n$ Terme, so ist $\underline{p}(\underline{t}_1,...,\underline{t}_n)$ eine Atomformel.

Beispiele für Atomformeln sind:
maennlich(a)
liebt(X,Y)
schreibt(verfasser_von(ulysses),buch). Variablenfreie Atomformeln stehen in der Umgangssprache für einfache Sätze ohne logische Verknüpfungen wie bei „Der Verfasser von Ulysses schreibt ein Buch". Es sei angemerkt, daß in Prolog syntaktisch nicht unterschieden wird zwischen Atomformeln und Termen als Funktoren mit Argumenten; verfasser_von(ulysses) ist syntaktisch auch eine Atomformel.
Schließlich können aus Atomformeln mit den logischen Konstanten und Quantoren komplexe Formeln gebaut werden:

(F1) Ist $\underline{F}$ eine Atomformel, so ist $\underline{F}$ auch eine Formel.
(F2) Sind $\underline{F}$ und $\underline{G}$ Formeln, so sind auch die folgenden Ausdrücke Formeln (mit abnehmender Bindungsstärke der logischen Konstanten in dieser Reihenfolge):
¬$\underline{F}$ (Negation), $\underline{F}$&$\underline{G}$ (Konjunktion), $\underline{F}$∨$\underline{G}$ (Disjunktion), $\underline{F}$=>$\underline{G}$ (Implikation), $\underline{F}$<=>$\underline{G}$ (Äquivalenz).
(F3) Ist $\underline{F}$ eine Formel und $\underline{X}$ eine Variable, so sind auch ∀$\underline{X}$:$\underline{F}$ und ∃$\underline{X}$:$\underline{F}$ Formeln.

∀$\underline{X}$ ist der Allquantor, ∃$\underline{X}$ der Existenzquantor. Formeln der Gestalt (F3) heißen All- bzw. Existenzformeln.
Für die unterstrichenen Symbole sind dabei wieder zulässige Ausdrücke der Prädikatenlogik einzusetzen. Es sei angemerkt, daß sich die Aussagenlogik als der Teil der Prädikatenlogik definieren läßt, der keine Funktionskonstanten, keine Individuenkonstanten und nur 0stellige Prädikatsymbole enthält (n = 0 in Definition (A)).

Ein Beispiel

Es soll nun an einem kleinen Beispiel veranschaulicht werden, wie man komplexe Formeln aufbaut. X und Y seien Variable und somit auch Terme, maennlich, kind_von und vater_von seien Prädikate. Nach (A) sind dann maennlich(X), kind_von(Y,X), vater_von(X,Y) Atomformeln. Wendet man (F2) an, so ergibt sich die komplexe Formel maennlich(X) & kind_von(Y,X) und bei nochmaliger Anwendung von (F2)
maennlich(X) & kind_von(Y,X) =>
vater_von(X,Y).
In dieser Formel kommen alle Variablen frei vor, d. h. sie sind an keinen Quantor gebunden. Wendet man mit (F3) den All-

quantor an, so erhält man eine geschlossene Formel, in der alle Variablen durch die Quantoren gebunden sind:

∀X:∀Y: maennlich(X) & kind_von(Y,X)
=> vater_von(X,Y).

Allgemein nennt man eine Variable gebunden, wenn sie im Wirkungsbereich eines Quantors liegt. Der Wirkungsbereich eines Quantors erstreckt sich dabei auf jene Formeln, deren Präfix er ist. Nur geschlossene Formeln ohne freie Variablen können wahr oder falsch sein. Die obige Formel kann natürlich mit weiteren Formeln zu noch komplexeren verknüpft werden.
Nimmt man zu der Allformel die beiden eingangs aufgeführten Fakten hinzu, so läßt sich z. B. logisch ableiten:

vater_von(leopold_mozart,
wolfgang_amadeus_mozart).

Wie lassen sich nun solche Schlüsse rechtfertigen?
Prinzipiell sind hierzu zwei Vorgehensweisen möglich:
- Entweder, man stellt einen sog. Kalkül mit einer Menge von Regeln auf und leitet aus den vorliegenden Formeln das zu beweisende Theorem mit den Regeln ab. Ein Beispiel für eine einfache Regel, die in den meisten Logikkalkülen Verwendung findet, ist der Modus Ponens, der besagt: aus F und F=>G darf G abgeleitet werden.
- Eine andere Möglichkeit besteht darin, einen Beweis semantisch zu führen, d. h. zu zeigen, daß, wenn die vorliegenden Formeln wahr sind, auch der Schluß zwingend wahr sein muß.

Bei der zweiten Möglichkeiten, auf die wir uns hier beschränken, muß geklärt werden, was es heißt: „eine Formel ist wahr". Hierzu müssen den Grundausdrücken Bedeutungen zugeordnet werden. Bislang sind die eingeführten Ausdrücke ja Zeichenketten ohne Bedeutung. Weder die zulässigen Terme wolfgang_amadeus_mozart noch xcfsfaf haben innerhalb der Prädikatenlogik eine Bedeutung.
Daß wolfgang_amadeus_mozart für uns mit einem bestimmten Menschen verbunden ist, liegt nur daran, daß diese Zeichenfolge in der Umgangssprache auf einen bekannten Komponisten verweist. Wir haben bislang immer stillschweigend angenommen, daß Prädikate und Konstanten auf ihre umgangssprachlichen Äquivalente verweisen, ohne eigentlich eine Rechtfertigung dafür zu haben. In der Semantik werden die Ausdrücke der Sprache über die Begriffe Interpretation und Modell auf bestimmte mengentheoretische Strukturen abgebildet.

Was eine Interpretation ist

Eine Interpretation ordnet Individuenkonstanten (z. B. sokrates oder xcfsfaf) Individuen der realen Welt zu (z. B. der Konstanten wolfgang_amadeus_mozart den konkreten Menschen W. A. Mozart), Prädikaten (geordnete) Mengen von Individuen (z. B. dem Prädikat ist_menschlich die Menge aller Menschen, dem 2stelligen Prädikat toetet eine Menge von geordneten Paaren aus Mördern und Ermordeten etc.).
Formal besteht eine Interpretation aus einem (nicht-leeren) Definitionsbereich D, über den die Variablen laufen und einer Funktion I, die

(1) den Individuenkonstanten Elemente aus D zuordnet:
a :→ I(a), I(a) ∈ D
für alle Konstanten a;
(2) n-stelligen Funktoren n-Tupel von Elementen aus D zuordnet:
f :→ I(f), I(f) ⊆ D^n für alle Funktoren f
⊆ bezeichne die Teilmengenrelation, D^n das n-fache Cartesische Produkt von D);
(3) n-stelligen Prädikaten n-Tupel von Elementen aus D zuordnet:
p :→ I(p), I(p) ⊆ D^n für alle Prädikatsymbole p.
(4) Atomformeln Wahrheitswerte zuordnet:
F :→ I(F), I(F) ∈ {wahr,falsch} für alle Atomformeln F.

Mit Hilfe dieser Festlegungen kann nun der Wahrheitsbegriff für Atomformeln definiert werden; Atomformeln sind die kleinsten Einheiten, denen Wahrheitswerte zukommen:
Unter einer Interpretation I ist eine Atomformel p(a_1,...,a_n) genau dann wahr, wenn gilt:
<I(a_1),...,I(a_n)> ∈ I(p).
Beispiel: toetet(kain,abel) ist wahr unter folgender Interpretation:
(i1) I(kain) = Kain
Dabei ist kain eine Zeichenfolge der Sprache, die vor der Interpretation ohne Bedeutung ist und Kain der konkrete Mensch.
(i2) I(abel) = Abel
(i3) I(toetet) = {...<Kain,Abel>, ...}
Dem Ausdruck toetet ordnet die Interpretation eine Menge von geordneten Paaren zu, wobei das erste Element der Mörder, das zweite der Ermordete ist. Diese Menge enthält u.a. auch das Paar <Kain,Abel>, nicht aber <Abel,Kain>.
Dann ist toetet(kain,abel) wahr, da <Kain, Abel> in der Menge I(toetet) enthalten ist.
Die Wahrheitswerte komplexer Formeln werden aus den Basisausdrücken berechnet und sind wie folgt festgelegt
(„gdw" ist die Abkürzung für „genau dann wenn"):
F&G ist wahr unter I gdw I(F) = I(G) = wahr.
F∨G ist wahr unter I gdw I(F) = wahr oder I(G)=wahr.
¬F ist wahr unter I gdw I(F) = falsch.
F=>G ist falsch unter I gdw I(F) = falsch.
F <=> G ist wahr unter I gdw I(F) = I(G).
∃X:F ist wahr unter I gdw es existiert ein d ∈ D so daß bei X/dIF wahr ist (X/d steht für: Ersetze alle Vorkommnisse von X durch d).
∀X:F ist wahr unter I gdw für alle d ∈ D gilt: F ist wahr bei X/d.
Fügt man zu den obigen Interpretationen noch hinzu:
(i4) I(bruder_von) =
{... <Kain,Abel>,
<Abel,Kain>,...},
so ist
toetet(kain,abel) & bruder_von(kain,abel)
wahr. Diese Formel wäre aber nicht wahr unter der Interpretation:
I(kain) = Abel
I(abel) = Kain;
Man nennt alle Interpretationen, die eine gegebene geschlossene Formel wahr machen, ein Modell dieser Formel. Beispielsweise ist die Interpretation (i1)-(i4) ein Modell für die obige konjunktiv verknüpfte Formel. Hat eine Formel mindestens ein Modell, d. h. gibt es mindestens eine Interpretation, die die Formel wahr macht, so ist sie erfüllbar.
Formeln, die nicht erfüllbar sind, sind widerspruchsvoll, z. B. ist
∃X: maennlich(X) & ¬ maennlich(X),
(die Formel behauptet, es gibt jemanden, der männnlich und nicht männlich ist) unter keiner Interpretation erfüllbar und somit widerspruchsvoll.
Schließlich läßt sich der logisch-sematische Folgerungsbegriff wie folgt festlegen: eine Formel F folgt aus einer Formelmenge M, genau dann, wenn unter allen Interpretationen, die M wahr machen, auch F immer wahr ist.
Wir haben somit die für unsere Belange wichtigsten Konzepte der Prädikatenlogik in einer freilich sehr lückenhaften und für einen Logiker recht oberflächlichen Weise vorgestellt, die uns in diesem Rahmen aber genügen sollen. Die Prädikatenlogik in der bislang aufgebauten üblichen Syntax eignet sich u. a. wegen Ineffizienz schlecht für Computerübertragung; alle Versuche, Logik als Computersprache zu benutzen mündeten in eine spezielle syntaktische Variante der Prädikatenlogik. Der zweite und letzte Teil dieses Beitrags im nächsten Heft beginnt deshalb mit der Clausenlogik.

Hagen Völzke

Digitale Bilderzeugung und -verarbeitung

Teil 3: Architektur von Grafikprozessoren

Im ersten und zweiten Teil dieser Artikelserie wurden der Aufbau eines Bildsignales, Zeitforderungen und einige konkrete Beispielschaltungen besprochen. In diesem Teil werden nun verschiedene Grafikprozessoren vorgestellt: Vom betagten 6845 bis hin zu ACRTC und GDP.

Auf den CRTC-Baustein 6845 wurde bereits im zweiten Teil kurz eingegangen. Er soll hier nur der Vollständigkeit halber erwähnt werden, da er eigentlich kein richtiger Grafikprozessor ist. Der CRTC 6845 wurde ursprünglich für zeichenorientierte Bildschirme entwickelt. Sein typisches Einsatzgebiet ist z. B. in einem Terminal. Der Baustein erzeugt die nötigen Speicheradressen, generiert einen vertikalen und einen horizontalen Synchronimpuls sowie das Austastsignal (Blank-Signal). Das Anzeigeformat und die resultierenden Ablenkfrequenzen lassen sich in weiten Bereichen einstellen, da der Baustein voll programmierbar ist.

Der CRTC 6845

Der Baustein kann wahlweise mit oder ohne Zeilensprung (interlaced/noninterlaced) arbeiten. Er enthält ferner einen programmierbaren Cursor, der in verschiedenen Betriebsarten dargestellt werden kann, u. a. Blinken (langsam/schnell), dauernd

$\overline{CS}$	RS	Address Register					Register	Register Name	Program Unit	READ	WRITE	Data Bit							
		4	3	2	1	0	#					7	6	5	4	3	2	1	0
1	×	×	×	×	×	×			–	–	–								
0	0	×	×	×	×	×	AR	Address Register	–	×	○								
0	1	0	0	0	0	0	R0	Horizontal Total *	Character	×	○								
0	1	0	0	0	0	1	R1	Horizontal Displayed	Character	×	○								
0	1	0	0	0	1	0	R2	Horizontal Sync* Position	Character	×	○								
0	1	0	0	0	1	1	R3	Horizontal Sync Width	Character	×	○								
0	1	0	0	1	0	0	R4	Vertical Total *	Line	×	○								
0	1	0	0	1	0	1	R5	Vertical Total Adjust	Raster	×	○								
0	1	0	0	1	1	0	R6	Vertical Displayed	Line	×	○								
0	1	0	0	1	1	1	R7	Vertical Sync * Position	Line	×	○								
0	1	0	1	0	0	0	R8	Interlace mode	–	×	○							V	S
0	1	0	1	0	0	1	R9	Maximum Raster Address	Raster	×	○								
0	1	0	1	0	1	0	R10	Cursor Start Raster	Raster	×	○		B	P					
0	1	0	1	0	1	1	R11	Cursor End Raster	Raster	×	○								
0	1	0	1	1	0	0	R12	Start Address(H)	–	×	○								
0	1	0	1	1	0	1	R13	Start Address(L)	–	×	○								
0	1	0	1	1	1	0	R14	Cursor(H)	–	○	○								
0	1	0	1	1	1	1	R15	Cursor (L)	–	○	○								
0	1	1	0	0	0	0	R16	Light Pen(H)	–	○	×								
0	1	1	0	0	0	1	R17	Light Pen(L)	–	○	×								

Bild 1. Die Register des CRTC 6845

an/aus, Block oder Linie. X- und Y-Position des Cursors lassen sich über zwei Register programmieren. Da der Baustein über eine einfache Busschnittstelle zur Haupt-CPU verfügt, läßt er sich mit geringem Aufwand in nahezu alle Computersysteme integrieren. Auch wegen seines niedrigen Preises und der guten Lieferbarkeit (viele Hersteller) ist dieser Baustein ziemlich stark verbreitet. Er wird häufig auch für reine Grafik verwendet, wobei man allerdings nur in wenigen Spezialfällen eine lineare Adreßverteilung erhält. Wegen seiner geringen Komplexität hält sich auch die Zahl seiner Register in Grenzen: *Bild 1* zeigt die internen Register des CRTC 6845, *Bild 2* gibt Aufschluß über deren Wirkung. In *Bild 3* schließlich finden Sie ein Blockschaltbild für den CRTC 6845.

Wird eine lineare Adreßverteilung gewünscht, so läßt sich das nur bei einer horizontalen Auflösung realisieren, die eine Potenz von 2 ist, z. B. 512 oder 1024 Punkte. Bei einer Auflösung von 640 oder 768 Punkten ergeben sich sehr unschöne Adreßsprünge. Allerdings kann man diese durch ein Umrechnungs-EPROM korrigieren: Das EPROM wird einfach in die Adreßleitungen des CRTC 6845 geschleift. Im EPROM steht dann jeweils die Umsetzinformation für die anliegende Adresse, mit der dann der Bildspeicher adressiert wird. Mit diesem Verfahren lassen sich allerdings nur fest gewählte Bildformate realisieren, was aber wohl in den meisten Fällen ausreicht.

Der CRTC 6845 eignet sich hauptsächlich für Grafikanwendungen, bei denen die CPU ohnehin den gesamten Bildaufbau übernehmen muß. Dies ist z. B. der Fall bei Bildverarbeitung, einfachen Grafikanwendungen ohne die Forderung nach hoher Zeichengeschwindigkeit, aber auch bei sehr komplexen Anwendungen wie etwa der Unterstützung von GKS. GKS heißt „Grafisches-Kern-System“ und ist ein sehr abstrakter und aufwendiger Standard für Computergrafik. Da praktisch alle Zeichenoperationen mit komplizierten Attributen beeinflußt werden können (Farbe, Muster, Strichbreite), waren auch moderne Grafikprozessoren bis vor kurzem hierfür nicht geeignet, da deren eingebaute Zeichenoperationen nur wenige einfache Spezialfälle der GKS-Norm abdekken können.

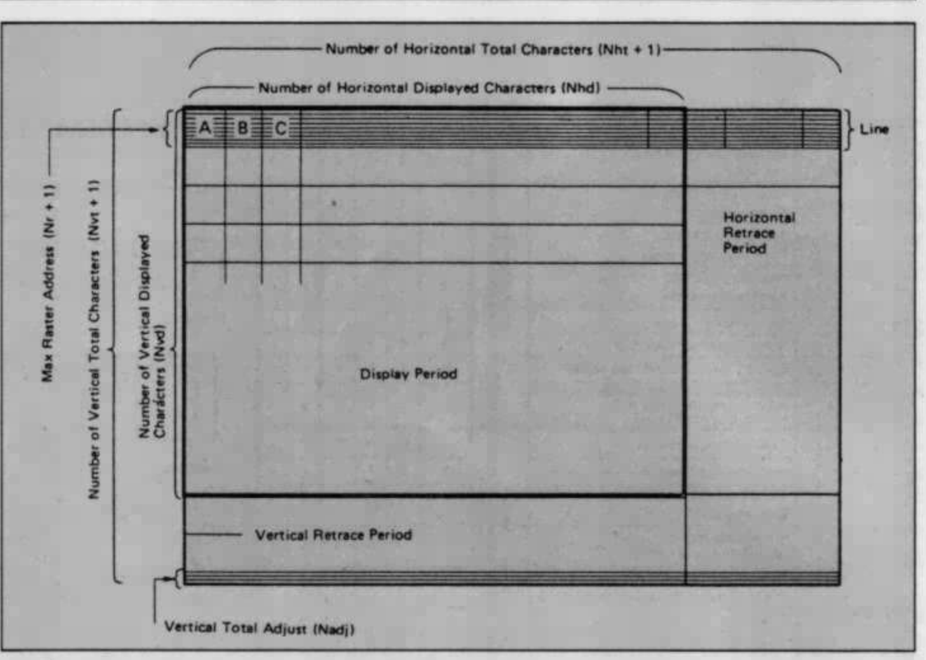


Bild 2. So sieht das Anzeigeformat des CRTC 6845 aus

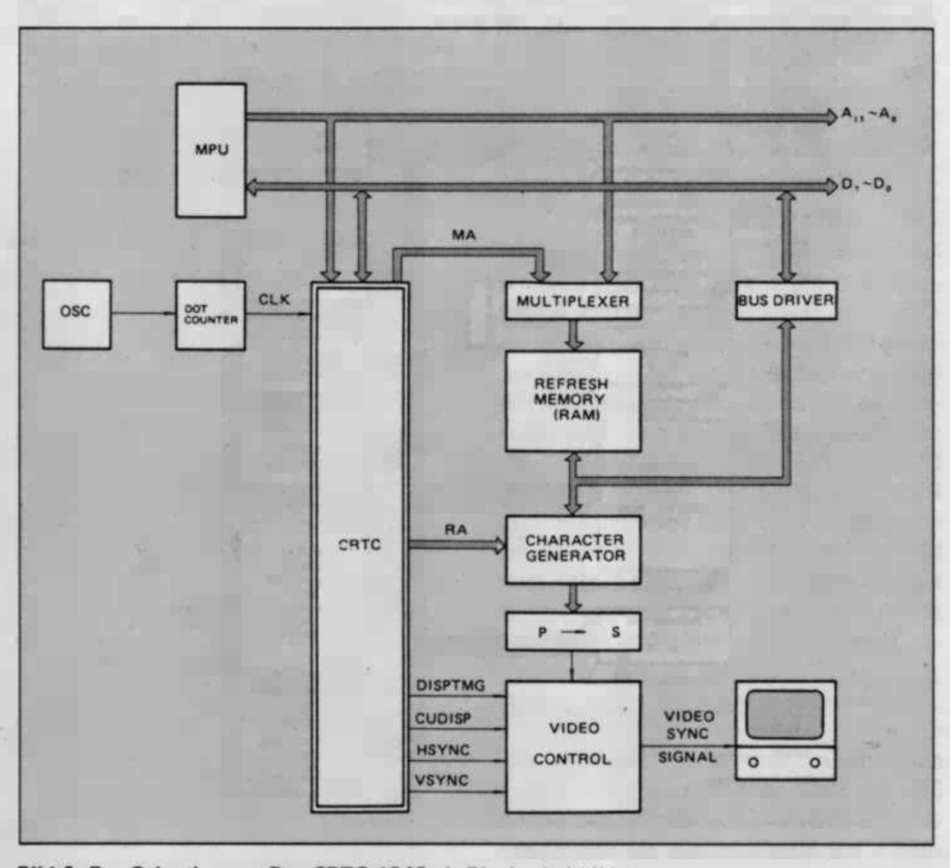


Bild 3. Zur Orientierung: Der CRTC 6845 als Blockschaltbild

Die Controller EF 9365 und EF 9366

Schon etwas komfortabler sind die von Thomson hergestellten Grafik-Controller EF 9365 und EF 9366. Die beiden Bausteine unterscheiden sich durch ihre Auflösung: Der EF 9365 gestattet wahlweise 512 × 512 Punkte (interlaced), 256 × 256, 128 × 128 oder 64 × 64 Punkte Auflösung. Der EF 9366 hat nur ein festes Bildformat von 512 × 256 Punkten (noninterlaced). Hier ist auch schon ein großer Nachteil dieser Bausteine ersichtlich: Sie gestatten nur feste Bildformate im unteren Auflösungsbereich. 512 Punkte horizontal sind nicht mehr Stand der Technik. Dafür verfügen Sie aber über einen eingebauten Prozessor, der mit sehr hoher Geschwindigkeit Linien und Zeichen erzeugen kann. Für die Zeichendarstellung verfügt der Baustein über einen eingebauten Zeichengenerator. Alle Zeichen können stehend, liegend oder wahlweise nach links oder rechts geneigt dargestellt werden. Durch die kleine Matrix von 5 × 7 Punkten lassen sich bis zu 87

b3 b2 b1 b0 \ b7 b6 b5 b4		0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0 … 1 1 1 1
		0	1	2	3	4	5	6	7	8 9 A B C D E F
0 0 0 0	0	Set bit 1 of CTRL1 : Pen select	Vector generation (for b2, b1, b0 see small vector definition)	SPACE	0	@	P	`	p	SMALL VECTOR DEFINITION :
0 0 0 1	1	Clear bit 1 of CTRL 1 : Eraser select		!	1	A	Q	a	q	
0 0 1 0	2	Set bit 0 of CTRL1 : Pen/Eraser down select		"	2	B	R	b	r	
0 0 1 1	3	Clear bit 0 of CTRL 1 : Pen/Eraser up select		#	3	C	S	c	s	
0 1 0 0	4	Clear screen		$	4	D	T	d	t	
0 1 0 1	5	X and Y registers reset to 0		%	5	E	U	e	u	
0 1 1 0	6	X and Y reset to 0 and clear screen		&	6	F	V	f	v	
0 1 1 1	7	Clear screen, set CSIZE to code "minsize" All other registers reset to 0 (except XLP, YLP)		'	7	G	W	g	w	
1 0 0 0	8	Light-pen initialization	Special direction vectors (for b2, b1, b0 see small vector definition)	(	8	H	X	h	x	
1 0 0 1	9	Light-pen initialization		)	9	I	Y	i	y	
1 0 1 0	A	5 x 8 block drawing (size according to CSIZE)		*	:	J	Z	j	z	
1 0 1 1	B	4 x 4 block drawing (size according to CSIZE)		+	;	K	[	k	{	
1 1 0 0	C	Screen scanning : Pen or Eraser as defined by CTRL1		,	<	L	\	l	¦	
1 1 0 1	D	X registers reset to 0		–	=	M	]	m	}	
1 1 1 0	E	Y registers reset to 0		.	>	N	↑	n	‾	
1 1 1 1	F	Direct image memory access request for the next free cycle.		/	?	O	←	o	▒	

SMALL VECTOR DEFINITION :

b7	b6 b5	b4 b3	b2 b1 b0
1	|ΔX|	|ΔY|	Direction

Dimension

ΔX or ΔY		Vector length
0	0	0 step
0	1	1 step
1	0	2 steps
1	1	3 steps

Direction

010, 001, 000, 101, 100, 111, 110, 011

Bild 4. Diese Befehle kennt der EF 9365/EF 9366

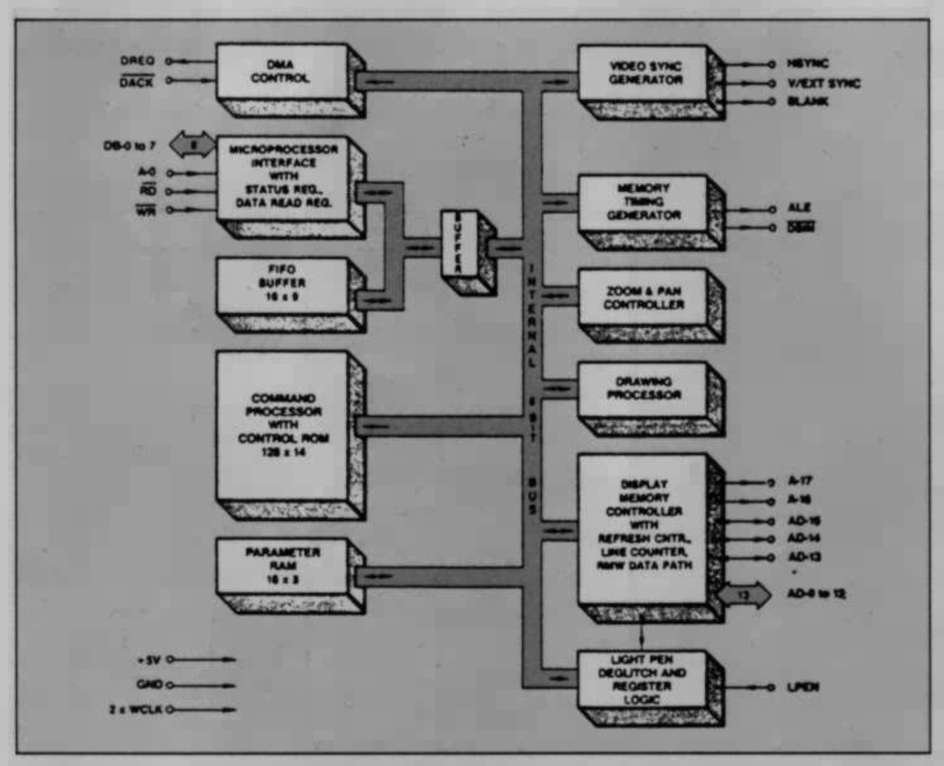


Bild 5. Das Blockdiagramm des NEC 7220

Zeichen pro Zeile darstellen. Der Controller erzeugt sämtliche Adreßinformationen zur Ansteuerung des Bildspeichers und die für den Monitor benötigten Blank- und Synchronsignale. Über ein eigenes Kommando gestattet er auch der Haupt-CPU den Zugriff auf den Bildspeicher. Der Baustein kann wahlweise für monochrome und farbige Applikationen verwendet werden. *Bild 4* zeigt einen Auszug aus dem Datenblatt der EF-Controller. Die verfügbaren Zeichenkommandos sind dort tabellarisch aufgeführt.

Der Grafikprozessor NEC 7220

Ein weit verbreiteter Grafikprozessor ist der NEC 7220. Ähnlich wie beim 6845 lassen sich verschiedene Bildformate programmieren, zusätzlich verfügt der Baustein über einen eingebauten Zeichenprozessor, der Linien, Kreise, Bogenstücke, Rechtecke und Zeichen erzeugen kann. Außerdem gibt es ein Fill-Kommando zum Ausfüllen einzelner Flächen. Die Steuerung des Bausteins erfolgt über eingebaute FIFO Speicher (First-In-First-Out), so daß mehrere Befehle auf einmal übergeben werden können und dann nacheinander im erforderlichen Zeitmaß vom Grafikprozessor abgearbeitet werden.

Der NEC 7220 unterstützt bereits Zooming und Panning sowie einen Cursor. Die Auflösung kann maximal 1024 x 1024 Punkte mit je 4 Bit betragen. DMA-Befehle gestatten die schnelle Übertragung von Daten aus dem Bildspeicher in den Hauptspeicher und umgekehrt.

Die Zeichendarstellung kann wahlweise per Software erfolgen, (Zeichenmatrizen werden gezeichnet) oder durch einen externen Zeichengenerator. Auch der NEC 7220 stellt die für den Monitor benötigten Blank- und Synchronsignale zur Verfügung. Das Businterface kommt dem Z80-Protokoll recht nahe; der Baustein läßt sich aber auch leicht in andere Systeme integrieren. *Bild 5* zeigt in einem Blockdiagramm den internen Aufbau des NEC 7220. Die Zeichenoperationen können wahlweise während der gesamten Zeit erfolgen (mit Störungen im Bild) oder nur in den Blanking-Phasen, wodurch natürlich die Zeichengeschwindigkeit stark herabgesetzt wird. Der Baustein kann wahlweise mit oder ohne Zeilensprung arbeiten.

Interleaving

An dieser Stelle soll noch ein neuer Begriff geklärt werden: „interleaving memory access“. Das heißt ungefähr „abwechselndes Zugreifen auf den Speicher“. Wie eben beim NEC 7220 geschildert, kann dieser Prozessor entweder nur in den Blanking-Phasen zeichnen, also dann, wenn keine Bildinformation aus dem Speicher gelesen werden muß, oder zu jedem Zeitpunkt, mit höherer Priorität als der Display-Prozessor. Letzteres hat zur Folge, daß bei jedem Zeichenzugriff ein Display-Zugriff ausfallen muß – an der entsprechenden Stelle auf dem Bildschirm erscheint eine falsche Bildinformation. Dies äußert sich als „Schnee“, der über das ganze Bild verteilt ist. Je schneller der Grafikprozessor zeichnet, desto dichter ist dieser Schnee. Der Vorteil ist natürlich eine hohe Zeichengeschwindigkeit. Für professionelle Anwendungen sind allerdings die Störungen untragbar. Um trotzdem die volle Zeichengeschwindigkeit zu erreichen, kann man immer abwechselnd einen Speicherzugriff für den Bildauf-

bau (Display) und einen für den Zeichenprozessor vorsehen. Dieses Verfahren ist allerdings ziemlich aufwendig, da bei gleichbleibender Bildauflösung ein doppelt so schneller Speicher benötigt wird. Wer den ersten Teil dieses Artikels gelesen hat, weiß, wie kritisch gerade die Geschwindigkeit des Bildspeichers in eine Applikation eingeht.

Durch das Interleaving-Verfahren erhält man – bei entsprechendem Aufwand – sowohl ein einwandfreies Bild, als auch eine hohe Zeichengeschwindigkeit. Dummerweise verschenkt man viele mögliche Speicherzugriffe in den Phasen, in denen der Zeichenprozessor nichts zu tun hat. Man sollte daher das Interleaving-Verfahren nur einsetzen, wenn die hohe Zeichengeschwindigkeit ständig benötigt wird.

Der Texas Grafikprozessor TMS 34010

Ein relativ moderner Grafikprozessor wurde letztes Jahr von Texas Instruments vorgestellt. Eigentlich ist dieser Baustein ein kompletter Mikroprozessor – mit allen Befehlen, die für die Programmausführung nötig sind, also auch Kontrollstrukturen, arithmetischen und allgemeinen Instruktionen. Zusätzlich verfügt der Prozessor über eine große Menge von Grafikbefehlen, die wie „normale" Befehle im Programmtext eingebunden werden. Für diesen Prozessor gibt es sogar einen C-Compiler. *Bild 6* zeigt einen Überblick über den Befehlsvorrat des TMS 34010. Intern verfügt der TMS 34010 über 31 universelle 32-Bit-Register, einen 32-Bit Mikroprozessor, einen 256 Byte großen Cache, eine ALU, Zusatzlogik zur Pixelverarbeitung, sowie einen CRT-Controller, der den Bildaufbau übernimmt.

Der Baustein kann mit Standard DRAMs, aber auch mit den neueren VRAMs arbeiten. VRAMs sind sogenannte Video-RAMs, die wie ein normales DRAM aufgebaut sind. Zusätzlich verfügen sie über ein eingebautes Schieberegister mit z. B. 256 Bit Länge. In einem Zugriff kann in das Schieberegister eine ganze „Reihe" (row) der Speichermatrix geladen und dann mit Taktfrequenzen bis rund 25 MHz seriell ausgegeben werden. Diese RAMs reduzieren ganz erheblich den Datendurchsatz im Bildspeicher, der durch den Bildaufbau entsteht. Allerdings sind einige Möglichkeiten (Hardwarefenster, Softscrolling und Zooming) nur sehr eingeschränkt möglich. Man nennt solche RAM-Bausteine auch häufig „dual ported RAM", da der Baustein über zwei voneinander unabhängige Ein-/Ausgänge (ports) benutzt werden kann.

Die Zeichengeschwindigkeit des TMS 34010 wird von Texas Instruments mit bis zu 48 Millionen Punkten/Sekunde angegeben, was sich allerdings wohl nur auf Blocktransfer-Befehle bezieht. Der maximale Befehlsdurchsatz liegt bei knapp 6 MIPS, entspricht also der Leistungsklasse des Prozessors 68020, allerdings werden kein virtueller Speicher oder komplexe Adressierungsarten unterstützt. Passend zu diesem Baustein wird eine Lookup-Table angeboten: die TMS 34070 Color Palette. Sie kann 16 aus 4096 Farben darstellen, unterstützt zwei Attribute und kann mit Punktfrequenzen bis zu 66 MHz betrieben werden. Mit dem TMS 34010 kann ein eigenständiges Computersystem aufgebaut werden, da kein zusätzlicher Mikroprozessor nötig ist.

Drawing Procedures		
Co-ordinates Conversion	MPU Software	MPU Software
Drawing Pre-process		
Drawing Process (Algorithms)		ACRTC
Drawing Execution		
Display Control	CRTC	
Synchronizing Signals Generation		
Others		

Bild 7. Aufgaben des ACRTC/MPU

Der ACRTC 63484

Wir kommen jetzt zum eigentlichen Schwerpunkt dieses Teils: Dem ACRTC 63484 von Hitachi. ACRTC steht für „Advanced-CRTC", also verbesserter CRTC. Der ACRTC kann als Nachfolger zum CRTC 6845 angesehen werden. Wie seine lange Nummer schon vermuten läßt, wurde er im Zuge der 68000-Peripherie entwickelt. Seine Busschnittstelle ist daher auch speziell auf die CPU 68000 abgestimmt und verfügt bereits über die asynchrone Handshake-Leitung $\overline{DTACK}$. Mit wenigen TTL-Gattern läßt sich der Baustein natürlich auch in Systeme mit anderen Prozessoren integrieren. Daß der Baustein nicht „68484" heißt,

GRAPHICS INSTRUCTIONS

ADDXY	RS,RD
CMPXY	RD,RD
CPW	RS,RD
CVXYL	RS,RD
DRAV	RS,RD
FILL	L
FILL	XY
LINE	Z
MOVX	RS,RD
MOVY	RS,RD
PIXBLT	B,L
PIXBLT	B,XY
PIXBLT	L,L
PIXBLT	L,XY
PIXBLT	XY,L
PIXBLT	XY,XY
PIXT	RS,*RD
PIXT	RS,*RD.XY
PIXT	*RS,RD
PIXT	*RS,*RD
PIXT	*RS.XY,RD
PIXT	*RS.XY,*RD.XY
SUBXY	RS,RD

MOVE INSTRUCTIONS

MOVB	RS,*RD
MOVB	*RS,RD
MOVB	*RS,*RD
MOVB	*RS,*RD(Disp)
MOVB	*RS(Disp),RD
MOVB	*RS(Disp),*RD(Disp)
MOVB	RS,@DAddress
MOVB	@SAddress,RD
MOVB	@SAddress,@DAddress
MOVE	RS,RD
MOVE	RS,*RD,F
MOVE	RS,-*RD,F
MOVE	RS,*RD+,F
MOVE	*RS,RD,F
MOVE	-*RS,RD,F
MOVE	*RS+,RD,F
MOVE	*RS,*RD,F
MOVE	-*RS,-*RD,F
MOVE	*RS+,*RD+,F
MOVE	RS,*RD(Disp),F
MOVE	*RS(Disp),RD,F
MOVE	*RS(Disp),*RD+,F
MOVE	*RS(Disp),*RD(Disp),F
MOVE	RS,@DAddress,F
MOVE	@SAddress,RD,F
MOVE	@SAddress,*RD+,F
MOVE	@SAddress,@DAddress,F

GENERAL INSTRUCTIONS

ABS	RD
ADD	RS,RD
ADDC	RS,RD
ADDI	IW,RD
ADDI	IL,RD
ADDK	K,RD
AND	RS,RD
ANDI	L,RD
ANDN	RS,RD
ANDNI	L,RD
BTST	K,RD
BTST	RS,RD
CLR	RD
CLRC	
CMP	RS,RD
CMPI	IW,RD
CMPI	IL,RD
DINT	
DIVS	RS,RD
DIVU	RS,RD
EINT	
EXGF	RD,F
LMO	RS,RD
MMFM	RS,List
MMTM	RS,List
MODS	RS,RD
MODU	RS,RD
MOVI	IW,RD
MOVI	IL,RD
MOVK	K,RD
MPYS	RS,RD
MPYU	RS,RD
NEG	RD
NEGB	RD
NOP	
NOT	RD
OR	RS,RD
ORI	L,RD
RL	K,RD
RL	RS,RD
SETC	
SETF	FS,FE,F
SEXT	RD,F
SLA	K,RD
SLA	RS,RD
SLL	K,RD
SLL	RS,RD
SRA	K,RD
SRA	RS,RD
SRL	K,RD
SRL	RS,RD
SUB	RS,RD
SUBB	RS,RD
SUBI	IW,RD
SUBI	IL,RD
SUBK	K,RD
XOR	RS,RD
XORI	L,RD
ZEXT	RD,F

PROGRAM CONTROL INSTRUCTIONS

CALL	RS
CALLA	Address
CALLR	Address
DSJ	RD,Address
DSJEQ	RD,Address
DSJNE	RD,Address
DSJS	RD,Address
EMU	
EXGPC	RD
GETPC	RD
GETST	RD
JAcc	Address
JRcc	Address
JRcc	Address
JUMP	RS
POPST	
PUSHST	
PUTST	RS
RETI	
RETS	[N]
TRAP	N

Bild 6. Der TMS 34010 besitzt den Befehlsvorrat eines kompletten Mikroprozessors

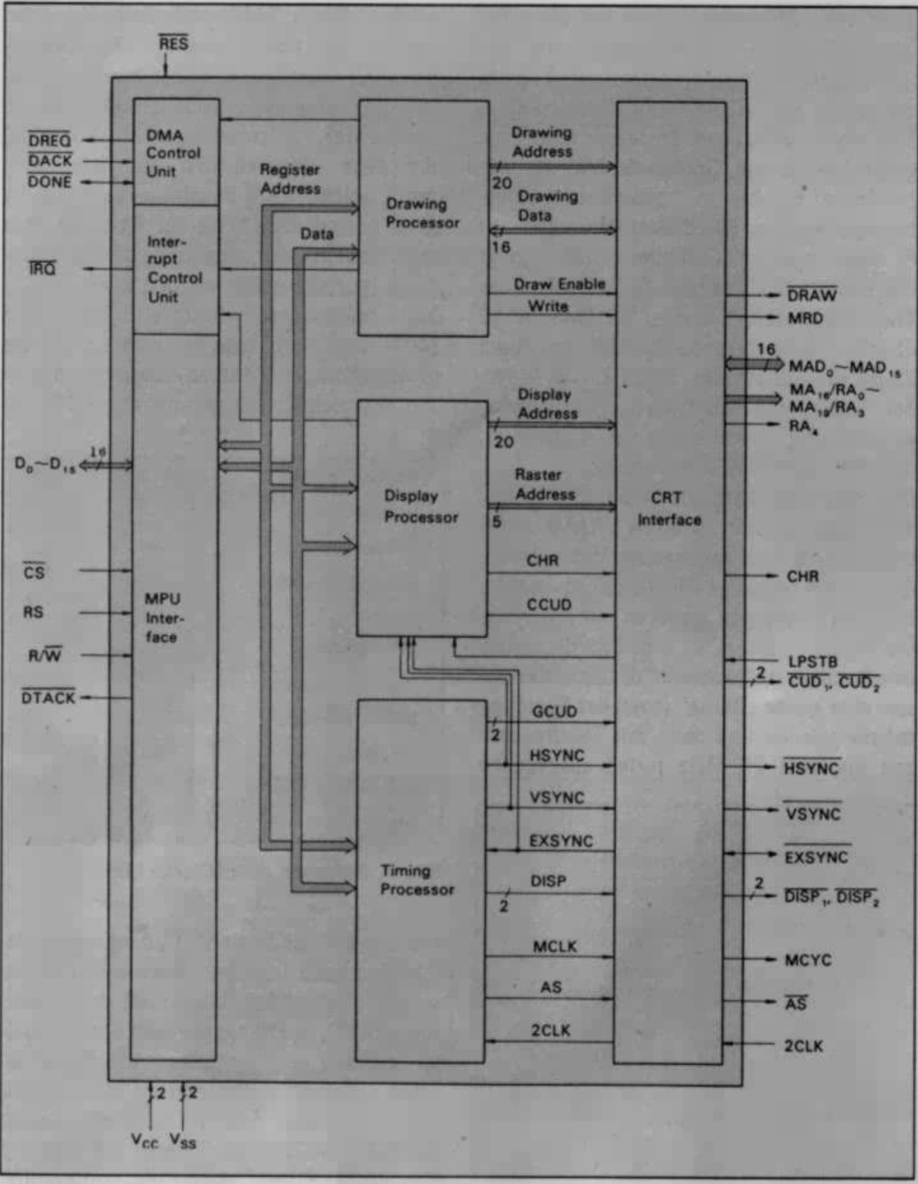


Bild 8. Der interne Aufbau des ACRTC

liegt daran, daß er in CMOS-Technik gefertigt ist und Hitachi normalerweise die „3" für CMOS und die „8" für NMOS-Technologie verwendet. Der ACRTC verfügt von allen vorgestellten Grafikprozessoren über den leistungsfähigsten Befehlssatz, was Zeichenoperationen betrifft. Er kann Punkte, Linien, Rechtecke, Linienzüge und Polygone, Kreise, Ellipsen und Bogenstücke zeichnen. Er verfügt über einen eingebauten PAINT-Befehl, der beliebig umrandete Flächen mit Farben und Mustern füllen kann. Block-Transfer-Befehle und Unterstützung für Zeichendarstellung runden seinen Befehlssatz ab.

Die erreichbare Zeichengeschwindigkeit ist sehr hoch, da der Baustein im Interleaving-Modus betrieben werden kann. Er unterstützt Speichertiefen von 1...16 Bit, so daß bis zu 65 536 verschiedene Farben pro Punkt dargestellt werden können. Die Adreßleitungen gestatten den Zugriff auf bis zu 2 MByte Grafikspeicher.

Der ACRTC unterstützt ein Hardware-Fenster, das frei positioniert werden kann. Unter Hinzunahme von externer Logik (siehe 2. Teil) können horizontales Softscrolling sowie horizontales Zooming realisiert werden. Interlacing ist natürlich auch möglich. In *Bild 7* ist dargestellt, welche Funktionen in einem Grafiksystem vom ACRTC übernommen werden können. *Bild 8* zeigt den internen Aufbau des ACRTC.

Die DMA Control Unit dient zur Abwicklung von direkten Speicherzugriffen des ACRTC auf den Hauptspeicher des Mikroprozessors. Hier können sehr schnell Daten aus dem Bildspeicher in den Hauptspeicher und umgekehrt übertragen werden. Ein externer DMA-Controller ist allerdings notwendig. Die Interrupt Control Unit ermöglicht das Erzeugen von Interrupts abhängig von 8 maskierbaren Bedingungen: Unzulässiger Befehl, Verlassen des Zeichenbereiches, Ende eines Befehls, Light Pen, Read-FIFO full, Read-FIFO empty, Write-FIFO ready und Write-FIFO empty. An den Interrupt-Bedingungen erkennt man schon, wie der Hauptprozessor mit dem ACRTC kommuniziert: Über einen Schreib- und einen Lese-FIFO. Damit sind wir auch schon beim MPU-Interface: Diese Schnittstelle gestattet den Zugriff auf die internen Kontrollregister des ACRTC und auf die beiden FIFO-Speicher.

Kein direkter Speicherzugriff

Ein direkter Zugriff auf den Bildspeicher ist nicht möglich, dieser muß über ein spezielles Kommando erfolgen. Die gewünschten Daten können dann dem Lese-FIFO-entnommen werden bzw. müssen vorher in den Schreib-FIFO gelegt werden. Gerade für komplexe Grafikstandards wie GKS oder auch GEM sowie für Bildverarbeitung ist aber ein direkter Zugriff auf den Bildspeicher wünschenswert. Hier muß der ACRTC passen. Man kann natürlich durch externe Zusatzhardware direkt auf den Bildspeicher zugreifen, muß aber schon ziemlich tricksen, um nicht mit Zugriffen des ACRTC in Konflikt zu geraten. Im übrigen gestatten auch die anderen Grafikprozessoren (EF 9365, NEC 7220, CRTC 6845) keinen direkten Zugriff auf den Bildspeicher. Beim CRTC 6845 ist eine Zusatzschaltung relativ einfach, da dieser Prozessor nur den Bildaufbau übernimmt. Bei den anderen Prozessoren gestaltet sich die ganze Sache ähnlich wie beim ACRTC schon schwieriger, da die Prozessoren auch Zeichenoperationen bzw. Refreshzyklen für dynamische Speicher durchführen. Lediglich der Texas Prozessor kann – konstruktionsbedingt – auf den Grafikspeicher direkt zugreifen. Der direkte Speicherzugriff ist nicht für alle Grafikanwendungen von Bedeutung, vereinfacht aber oft den Zugang zu den Bildinformationen (etwa das Abspeichern der Bitmap auf Diskette, Ausdrucken einer Hardcopy usw.).

Der Zeichen-Prozessor des ACRTC

Drei Komponenten innerhalb des ACRTC versorgen den Bildspeicher mit Signalen: Der Drawing-(= Zeichen-)Prozessor, der Display-Prozessor und der Timing-Prozessor. Die CRT-Schnittstelle bereitet diese Signale in geeigneter Weise auf und stellt sie dem Speicher zur Verfügung. Der Zeichenprozessor hat prinzipiell drei verschiedene Betriebsarten: Interleaving, non-interlea-

TYPE	MNEMONIC	COMMAND NAME
Register Access Command	ORG	Origin
	WPR	Write Parameter Register
	RPR	Read Parameter Register
	WPTN	Write Pattern RAM
	RPTN	Read Pattern RAM
Data Transfer Command	DRD	DMA Read
	DWT	DMA Write
	DMOD	DMA Modify
	RD	Read
	WT	Write
	MOD	Modify
	CLR	Clear
	SCLR	Selective Clear
	CPY	Copy
	SCPY	Selective Copy
Graphic Command	AMOVE	Absolute Move
	RMOVE	Relative Move
	ALINE	Absolute Line
	RLINE	Relative Line
	ARCT	Absolute Rectangle
	RRCT	Relative Rectangle
	APLL	Absolute Polyline
	RPLL	Relative Polyline
	APLG	Absolute Polygon
	RPLC	Relative Polygon
	CRCL	Circle
	ELPS	Ellipse
	AARC	Absolute Arc
	RARC	Relative Arc
	AEARC	Absolute Ellipse Arc
	REARC	Relative Ellipse Arc
	AFRCT	Absolute Filled Rectangle
	RFRCT	Relative Filled Rectangle
	PAINT	Paint
	DOT	Dot
	PTN	Pattern
	AGCPY	Absolute Graphic Copy
	RGCPY	Relative Graphic Copy

Bild 9. Der ACRTC hat den umfangreichsten Befehlsvorrat in bezug auf die Zeichenbefehle

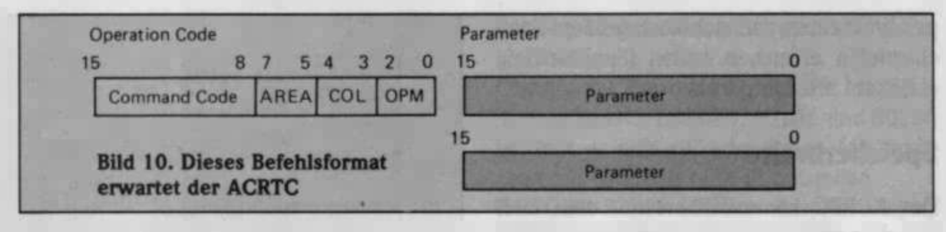


Bild 10. Dieses Befehlsformat erwartet der ACRTC

OPM			Operation Mode
0	0	0	REPLACE: Replaces the frame buffer data with the color data.
0	0	1	OR: ORs the frame buffer data with the color data. The result is rewritten to the frame buffer.
0	1	0	AND: ANDs the frame buffer data with the color data. The result is rewritten to the frame buffer.
0	1	1	EOR: EORs the frame buffer data with the color data. The result is rewritten to the frame buffer.
1	0	0	CONDITIONAL REPLACE (P=CCMP): When the frame buffer data at the drawing position (P) is equal to the comparison color (CCMP), the frame buffer data is replaced with the color data.
1	0	1	CONDITIONAL REPLACE (P≠CCMP): When the frame buffer data at the drawing position (P) is not equal to the comparison color (CCMP), the frame buffer data is replaced with the color data.
1	1	0	CONDITIONAL REPLACE (P<CL): When the frame buffer data at the drawing position (P) is less than the color register data (CL), the frame buffer data is replaced with the color data.
1	1	1	CONDITIONAL REPLACE (P>CL): When the frame buffer data at the drawing position (P) is greater than the color register data (CL), the frame buffer data is replaced with the color data.

Bild 11. Die möglichen Attribute für Zeichenoperationen

AREA			Drawing Area Mode
0	0	0	Drawing is executed without Area checking.
0	0	1	When attempting to exit the Area, drawing is stopped and the ARD (Area Detect) and CED (Command End) flags are set.
0	1	0	Drawing suppressed outside the Area – drawing operation continues and the ARD flag is not set.
0	1	1	Drawing suppressed outside the Area – drawing operation continues and the ARD flag is set.
1	0	0	Same as AREA = 0 0 0.
1	0	1	When attempting to enter the Area, drawing is stopped and the ARD and CED (Command End) flags are set.
1	1	0	Drawing suppressed inside the Area – drawing operation continues and the ARD flag is not set.
1	1	1	Drawing suppressed inside the Area – drawing operation continues and the ARD flag is set.

Bild 12. So werden die Bedingungen für die Clipping-Area gesetzt

ving flickering (mit Schnee) oder non-interleaving retrace (nur während Blank- und Rücklaufphasen). Durch die Implementation aller drei Betriebsarten wird der Baustein sehr flexibel, von Low-Cost-Applikationen bis hin zu High-End-Produkten einsetzbar. *Bild 9* zeigt den vollständigen Befehlssatz des Zeichenprozessors. Neben den vorher erwähnten Zeichenbefehlen finden sich noch DMA-Steuerbefehle, Positionier- und Initialisierungs-Befehle. *Bild 10* zeigt das allgemeine Befehlsformat des ACRTC.

Jede Zeichenoperation des ACRTC kann mit einem von acht möglichen Attributen versehen werden (*Bild 11*). Darunter befinden sich auch vier bedingte Attribute, die Veränderungen des Bildspeichers nur unter bestimmten Voraussetzungen vornehmen. Über zwei Bit im Befehlswort läßt sich die Farbzusammensetzung festlegen. Drei Bit gestatten die Verknüpfung des Befehles mit den Area-Checks (Bereichs-Prüfungen). *Bild 12* zeigt alle möglichen Kombinationen. Die „Area" ist ein rechteckiger Bereich, der oft auch „clipping rectangle" genannt wird (Abschneide-Rechteck). Das Clipping-Rectangle dient zum Abgrenzen einzelner Bildbereiche, beispielsweise von Fenstern. Jede Zeichenoperation kann vor ihrer Ausführung daraufhin überprüft werden, ob sie innerhalb des Clipping-Rectangle liegt. So kann man vermeiden, daß z.B. ein Kreisbogen aus einem Fenster herausläuft und andere Speicherbereiche beeinflußt. Schließlich soll ja keine Zeichenoperation aus einem Fenster herausführen. Durch diese dynamische Bereichsprüfung ist es nicht mehr Aufgabe der Steuersoftware, Koordinaten auf bestimmte Grenzen zu überprüfen: Bei einem Kreis wäre schon ein ziemlicher Rechenaufwand nötig. Außerdem besteht so die Möglichkeit, nur einen kleinen Ausschnitt einer größeren Zeichnung in einem Fenster anzuzeigen.

Der Display-Prozessor des ACRTC

Der Display-Prozessor erzeugt die (Refresh-) Adressen zum Bildaufbau sowie Informationen für die Erzeugung eines (oder auch zweier) Cursor. Er wird über eine ganze Menge von Registern programmiert, so sind Bildformat, Synchronimpulsdauer, Cursor-Position und Art (Muster/Fadenkreuz), Position und Größe des Hardware-Fensters sowie Zoom-Faktoren einstellbar. Der ACRTC unterstützt vier Bildbereiche (siehe *Bild 13*): Den sogenannten Base-Screen und zwei Split-Screens, den Upper- und den Lower-Split-Screen, sowie das Hardware-Fenster. Die Split-Screens erstrecken sich jeweils über die volle Bildbreite und können nur in ihrer Höhe verändert werden. Das Fenster kann beliebig positioniert werden, will man es horizontal punktweise verschieben können, so bedarf es allerdings wieder etwas Zusatzlogik, die das Softscrolling des Fensters übernimmt. Die beiden Split-Screens können beispielsweise für eine Menü- und eine Statuszeile verwendet werden, während auf dem Basis-Screen die eigentliche Grafik bearbeitet wird. Für jeden der vier Bildbereiche lassen sich die Adressen im Bildspeicher einzeln programmieren. Der Display-Prozessor kann wahlweise mit oder ohne Zeilensprung arbeiten. Durch die Möglichkeit der externen Synchronisation können in einem großen System mehrere ACRTCs miteinander gekop-

pelt werden, so daß sich noch größere Speichertiefen erreichen lassen (Speichertiefe = Bitzahl pro Bildpunkt).

Speicherbreite

Der ACRTC kommuniziert mit dem Bildspeicher über einen 16 Bit breiten Datenbus. Da aber für hohe Bildauflösungen aus Zeitgründen mehr als 16 Bit (auf Seite des Monitors) benötigt werden, unterstützt der ACRTC eine Speicherbreite bis zu 256 Bit. Externe Multiplexer sind notwendig, um die 16 Datenleitungen des ACRTC jeweils mit der richtigen Speicherbank zu verbinden. Übliche Applikationen mit dem ACRTC arbeiten fast immer mit einem 64 Bit breiten Speicher. Der ACRTC unterstützt die Verwendung überbreiter Speicher und beachtet dies bei seiner Adreßberechnung während Zeichenoperationen.
Außerdem verfügt der ACRTC über einen zusätzlichen Refresh-Generator, der für die für DRAMs nötigen Refreshzyklen sorgt. Prinzipiell reicht alleine der Bildaufbau zum Auffrischen der Speicher, jedoch kann

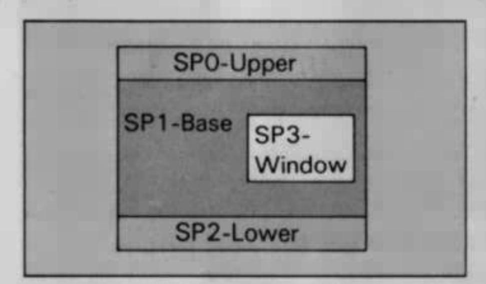


Bild 13. Die Bildschirmaufteilung beim ACRTC

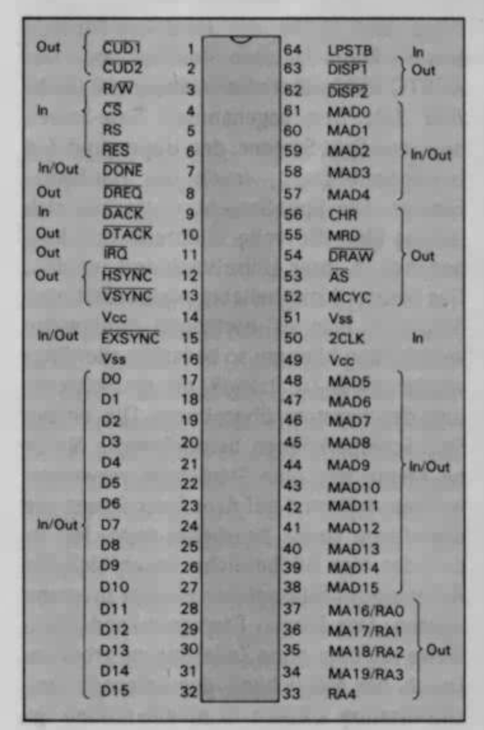


Bild 14. Pinbelegung beim ACRTC

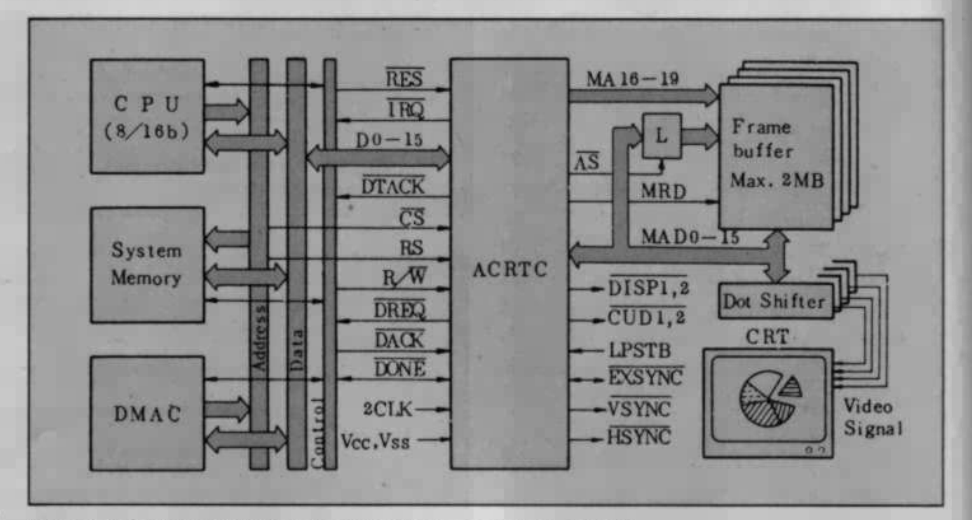


Bild 15. Blockschaltbild für eine Applikation mit dem ACRTC

dies nicht für alle Formate bzw. Kombinationen von Fenstern und Split-Screens garantiert werden. Daher ist dieser zusätzliche Refreshgenerator vorhanden, der übrigens immer in den horizontalen Synchronpausen arbeitet.

Der Timing-Prozessor

Aus den Daten des Display-Prozessors werden im Timing-Prozessor die Synchronimpulse, Blank-Signale sowie die Adreßsignale für den Speicher abgeleitet. Der ACRTC verfügt über zwei Blanksignale, wobei das eine für die Basis-Bildseite dient und das zweite für das Hardware-Fenster. *Bild 14* zeigt die Pinbelegung des ACRTC 63484 im 64poligen DIL-Gehäuse; inzwischen ist der Baustein auch im PLCC-Gehäuse erhältlich. Man sieht gleich, daß der Bus zum Bildspeicher gemultiplext ist. Man braucht daher externe Latches, die die Adressen zwischenspeichern. Insgesamt wird der Aufwand an externer Hardware recht groß, da folgende Funktionsgruppen extern realisiert werden:

- Adreßlatches
- Adreßmultiplexer für DRAMs und Puffer
- Datenmultiplexer zur Busbreiten-Anpassung
- Schieberegister für Punkterzeugung
- Takterzeugung für ACRTC
- Mischlogik für Cursor, Window, Blank- und Synchronsignale
- Zusatzlogik für horizontales Zooming und Soft-Scrolling
- Zusatzlogik für punktweises Setzen des Fadenkreuz-Cursors
- falls gewünscht: DMA-Controller

Dies bläht eine Applikation mit dem ACRTC ganz erheblich auf. Da es sich um komplexe und zeitkritische Funktionen handelt, lassen sie sich nicht in PAL-Bausteinen realisieren. Aus diesem Grund wurde von Hitachi zum ACRTC ein Chip-Satz entwickelt (Gate-Arrays), die alle Zusatzfunktionen (bis auf DMA) übernehmen. Das sind nochmals drei Bausteine im PLCC-Gehäuse. Neben dem ACRTC und diesen Bausteinen braucht man dann nur noch die RAMs und eventuell Bustreiber, um an den Systembus des Computersystems anzukoppeln. *Bild 15* zeigt ein Grob-Blockschaltbild für eine Applikation mit dem ACRTC, aus dem allerdings die eben geschilderte Komplexität nicht ersichtlich ist.

Alles in einem: Der GDP 82786

Noch relativ neu und daher recht unbekannt ist der von Intel entwickelte Grafikprozessor GDP 82786: Vielleicht der zur Zeit modernste und vielseitigste Grafikprozessor. Er enthält sämtliche Funktionen des ACRTC einschließlich des Chip-Satzes auf einem Baustein, zusätzlich einen eingebauten bidirektionalen DMA-Controller, der sowohl direkte Zugriffe des Hauptprozessors auf den Bildspeicher als auch Zugriffe des Grafikprozessors auf den Hauptspeicher erlaubt.
Der Displayprozessor wird nicht über interne Register programmiert, sondern über eine im Bildspeicher befindliche verkettete Liste (linked list) von Deskriptoren, die eine nahezu beliebige Anzahl von Hardwarefenstern ermöglichen. Dynamische Speicherbausteine lassen sich direkt an den Grafikkontroller anschließen, er liefert – ausreichend getrieben – gemultiplexte RAS/CAS-Adressen, RAS-, CAS-, Read-/Write- und Chip-Select-Signale.
Der Video-Ausgang ist geeignet, direkt einen Farbmonitor mit TTL-Eingang anzusteuern, kann jedoch ebenso über eine

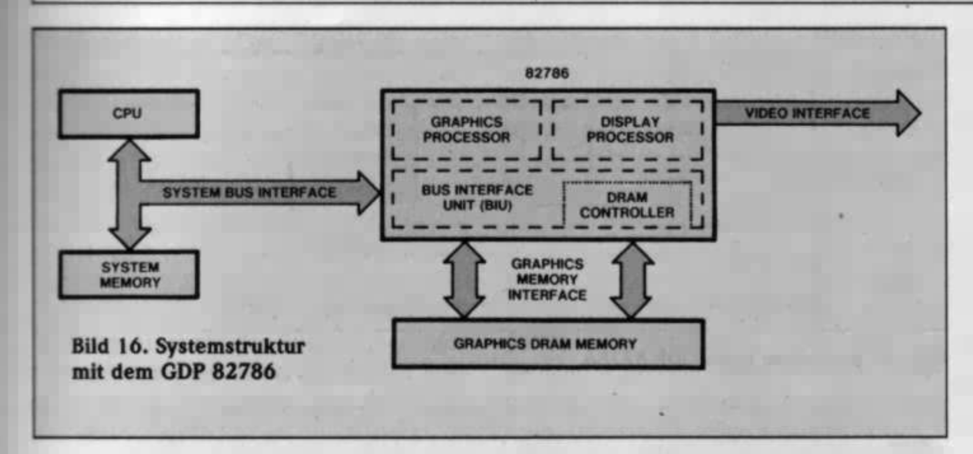


Bild 16. Systemstruktur mit dem GDP 82786

Command	Opcode	Command	Opcode
LINK	02	DEF_CHAR_ORIENT	4E
NOP	03	ABS_MOVE	4F
DEF_TEXTURE_OPAQUE	06	REL_MOVE	52
DEF_TEXTURE_TRANSPARENT	07	POINT	53
DEF_CHAR_SET_WORD	0A	LINE	54
DEF_CHAR_SET_BYTE	0B	RECT	58
INTR_GEN	0E	BIT_BLT	64
ENTER_MACRO	0F	ARC_EXCLUSION	68
EXIT_MACRO	17	ARC_INCLUSION	69
DEF_BIT_MAP	1A	POLYGON	73
DUMP_REG	29	POLYLINE	74
LOAD_REG	34	CIRCLE	8E
DEF_COLOR	3D	CHAR_OPAQUE	A6
DEF_LOGICAL_OP	41	CHAR_TRANSPARENT	A7
ENTER_PICK	44	BIT_BLT_M	AE
EXIT_PICK	45	INCR_POINT	B4
DEF_CLIP_RECT	46	HORIZ_LINES	BA
DEF_CHAR_SPACE	4D		

Bild 17. Befehlssatz des GDP 82786

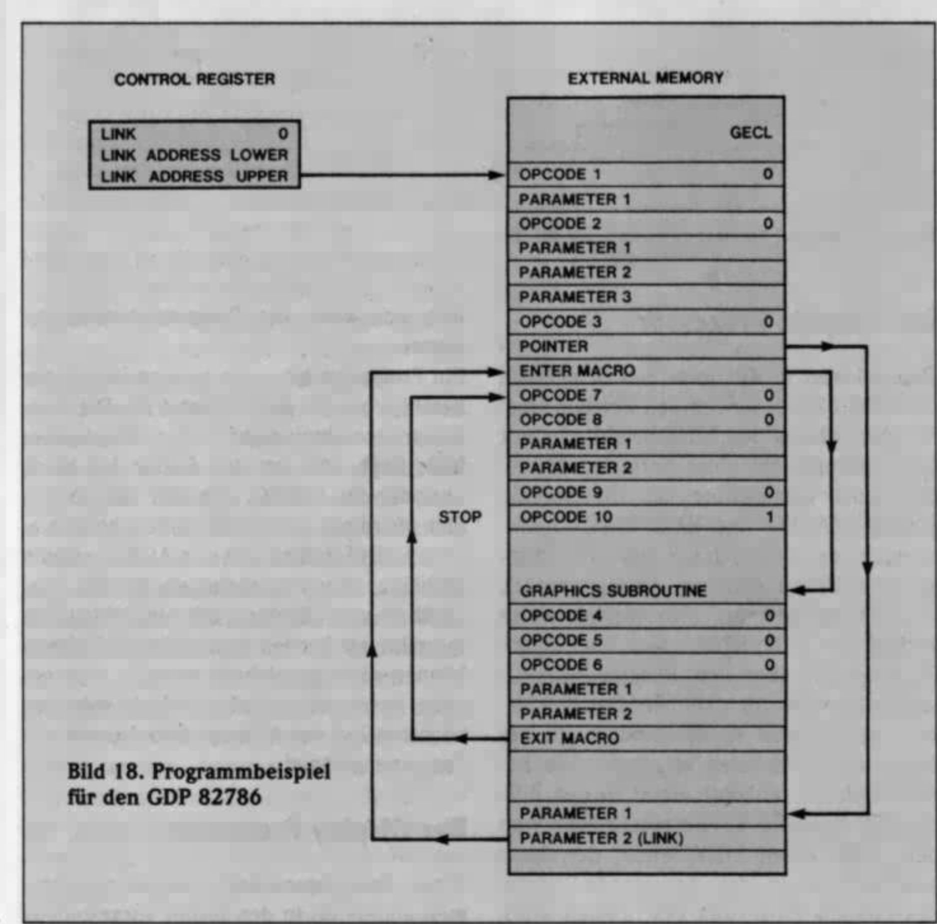


Bild 18. Programmbeispiel für den GDP 82786

Look-up-Table erweitert werden und analoge Farbsignale liefern.

Obwohl der Prozessor kompatibel zu Intels Mikroprozessoren 8086, 80186 und 80286 ist, läßt er sich auf asynchrone Mikroprozessoren (wie den 68000) umstellen.

Der GDP 82786 (GDP = Graphic Display Processor) verfügt über einen eigenen Befehlssatz, der programmartig im Speicher abgelegt werden kann. Es besteht sogar die Möglichkeit, Unterprogramme – Macros genannt – aufzurufen. Allerdings verfügt der Befehlssatz nicht über Ablaufstrukturen wie der des Texas-Prozessors TMS 34010.

Das Konzept der Speicherzugriffe auf den Bildspeicher ist vollkommen anders als beim ACRTC: Es wird nicht in festen Zeitscheiben zugegriffen, sondern immer nur bei Bedarf. Bildinformationen werden unter Ausnutzung der Seiten-(= Page-Mode-) Adressierung der DRAMs zunächst in einen eingebauten FIFO geladen und von dort mit der Punktfrequenz herausgeschoben.

Zugriffe des Zeichenprozessors und der Haupt-CPU werden jederzeit mit geringer Verzögerung zwischengeschoben. Die Priorität der einzelnen Zugriffe ist programmierbar.

Fenster können punktweise positioniert werden, ebenso sind horizontales Zooming sowie Soft-Scrolling implementiert. Jedes Fenster kann über eine eigene Bit-Tiefe verfügen, so daß Bereiche mit Text beispielsweise nur 1 Bit pro Punkt benötigen, Bereiche mit Grafik hingegen bis zu 8 Bit pro Punkt. Farben und Zooming lassen sich für jedes Fenster getrennt wählen. Außerdem erzeugt der Prozessor auf Wunsch hardwaremäßig einen Rahmen für jedes Fenster in einer wählbaren Farbe.

Der GDP 82786 unterstützt einen Hardware-Cursor, der wahlweise ein beliebiges 8 × 8- oder 16 × 16-Bit-Muster sein kann, oder ein Fadenkreuz über den ganzen Bildschirm. Der Cursor wird durch eine eingebaute Overlaytechnik erzeugt, so daß keine Veränderung am Bildspeicher nötig ist.

Die Zeichendarstellung ist äußerst flexibel: Jedes Zeichen besteht aus einer Matrix von bis zu 16 × 16 Bit, Proportionalschrift ist möglich, die Zeichensätze werden per Software geladen, somit lassen sich praktisch beliebig viele Zeichen darstellen.

Die externe Beschaltung dieses Bausteines ist extrem klein: Neben etwas Logik zur Ankopplung des Prozessors an das Bussystem des jeweiligen Computers wird nur noch der Speicher benötigt, eventuell eine Lookup-Table. Zwei getrennte Quarzoszillatoren versorgen den Baustein zum einen mit dem Grundtakt für die Speicherzugriffe und den Zeichenprozessor, zum anderen

mit der Punktfrequenz. Der Baustein ist für eine maximale Punktfrequenz von 25 MHz spezifiziert, die neuere Version sogar bis 35 MHz.

Der GDP 82786 unterstützt eine große Palette von Speicherbausteinen (durchweg DRAMs), z.B. 64 KBit, 256 KBit, 1 MBit, und VRAMs. Maximal können 4 MByte Speicher adressiert werden. Zugriffszeit und damit Preis der Speicher lassen sich ebenfalls programmieren, so daß je nach Anwendung und gewünschter Auflösung bzw. Bittiefe verschiedene Ausbaustufen bzw. Preisklassen einer Schaltung mit dem GDP möglich sind.

Bild 16 zeigt eine typische Systemstruktur bei Einsatz des 82786. Der Schaltungsaufwand ist tatsächlich so gering, wie das Blockschaltbild vermuten läßt. Eine detaillierte Schilderung über die Programmierung erscheint im nächsten Heft, wo auch eine Schaltung für eine Grafikkarte mit dem GDP 82786 für den mc-68000-Computer vorgestellt wird. Doch zunächst ein kurzer Überblick über die internen Baugruppen.

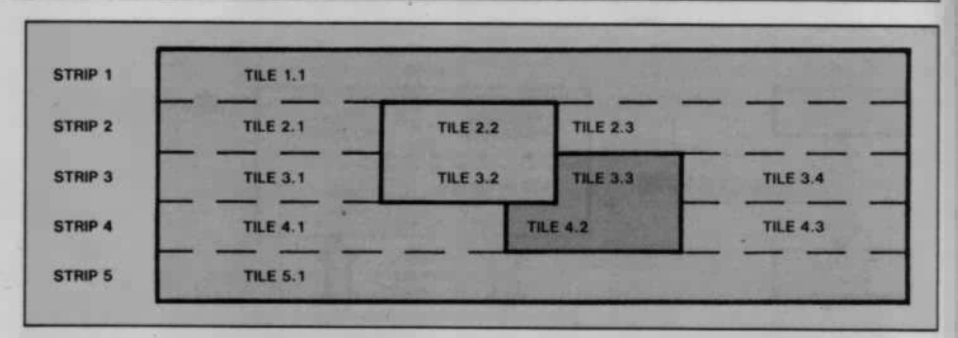


Bild 19. Bildaufbau beim GDP 82786

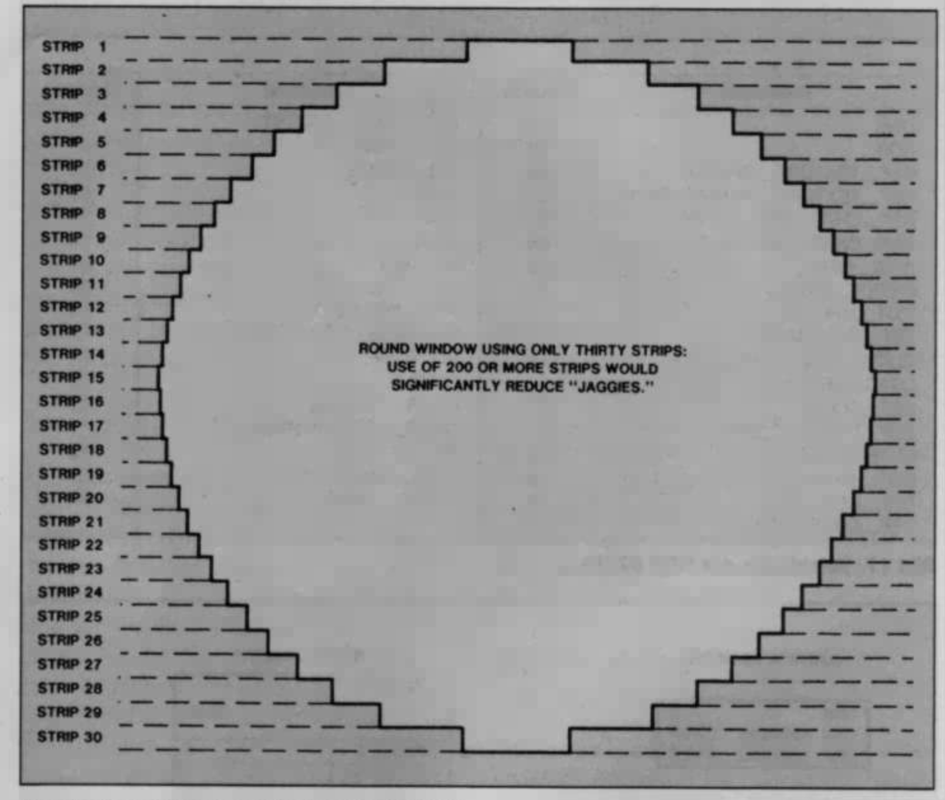


Bild 20. Beispiel-Fenster für den GDP 82786

Die Bus Interface Unit (BIU)

Die BIU verwaltet die Zugriffe der einzelnen Komponenten auf den Grafikspeicher. Zugreifen dürfen die Haupt-CPU, der Displayprozessor, der Grafikprozessor sowie der DRAM-Controller. Über drei Register läßt sich die Priorität der einzelnen Komponenten gegeinander einstellen. Die BIU gestattet der Haupt-CPU den direkten Zugriff auf den gesamten Bildspeicher. Die CPU legt dazu die Adressen an den GDP 82786 an und teilt ihm über Chip-Select mit, daß ein Zugriff stattfinden soll. Der GDP wiederum multiplext die anliegenden Adressen intern und steuert damit den Speicher an. Wenn die gewünschten Daten gelesen/geschrieben wurden, wird dies durch eine Handshake-Leitung angezeigt. Während des Zugriffes schaltet sich die Haupt-CPU mit Tristate-Puffern auf den Datenbus des Bildspeichers. Diese Puffer werden extern benötigt (2 ICs). Die BIU steuert auch den Zugriff auf die internen Register des GDP 82786.

Der DRAM Controller

Dieses Modul erzeugt $\overline{RAS}$, $\overline{CAS}$ und $\overline{\text{Chip-Select}}$-Signale für die DRAMs sowie Refresh-Adressen. Der DRAM-Controller ist für die unterschiedlichsten RAM-Bausteine konfigurierbar. Die Refresh-Frequenz ist ebenfalls programmierbar. Maximal können 4 MByte RAM adressiert werden. Der GDP 82786 kann bis zu 32 RAM-Bausteine ohne zusätzliche Treiber ansteuern.

Der Graphic-Prozessor

Bild 17 zeigt in Kurzform den Befehlssatz des GDP 82786. Neben den Zeichen-Operationen gibt es den LINK-Befehl, der mit dem Sprungbefehl eines normalen Mikroprozessors vergleichbar ist. Die Befehle ENTER–MACRO und EXIT–MACRO entsprechen den Befehlen JSR und RTS (Springe zum Unterprogramm, Springe zurück vom Unterprogramm). Notwendigerweise verfügt der GDP 82786 also über einen Stackpointer, über den Rücksprungadressen gesichert werden können. Der Befehlssatz enthält keine Kontrollbefehle, die bedingte Verzweigungen erlauben. Alle Befehle müssen zunächst direkt in den Bildspeicher des GDP 82786 geschrieben werden. Über einen LINK-Befehl, der daran anschließend in das Kommando-Register des Graphik-Prozessors geschrieben wird, läßt sich dann die Programmausführung starten.

Ein Programm ist zu Ende, wenn in einem Befehlswort das Bit 0 gesetzt ist. Die Ausführungsgeschwindigkeit der Zeichenbefehle liegt etwa um den Faktor 1,5 bis 2 über der des ACRTC. Der GDP 82786 verfügt allerdings nicht über einen komfortablen PAINT-Befehl wie der ACRTC. Dafür enthält er aber Vorkehrungen, die das „Anklicken" von Objekten mit einer Maus im interaktiven Betrieb unterstützen. Ellipsen können nicht gezeichnet werden. Eine genaue Beschreibung aller Befehle folgt im nächsten Teil des Artikels. *Bild 18* zeigt ein Programmbeispiel.

Der Display-Prozessor

Diese Funktionseinheit arbeitet vollkommen anders als in den bisher vorgestellten

Grafikprozessoren. Der Display-Prozessor wird zunächst über einen Steuerblock auf die Synchronfrequenzen, das Bildformat, die Cursorbetriebsart sowie Farbinformationen eingestellt. Der Steuerblock wird per Kommando aus dem Bildspeicher in die internen Register des GDP 82786 übertragen. Gleichzeitig wird in diesem Steuerblock die Anfangsadresse einer verketteten Liste übergeben. In dieser Liste stehen nun für jeden Abschnitt des Bildschirmes Informationen über die zu verwendenden Speicheradressen, die Bittiefe und die Größe dieses Abschnittes. Im einfachsten Fall überdeckt ein Abschnitt den gesamten Bildschirm. Im allgemeinen jedoch wird der Bildschirm in sogenannte Strips (Streifen) eingeteilt, die im Extremfall nur eine Linie hoch sein können (*Bild 19*). Jeder Streifen wird nochmals unterteilt in bis zu 16 horizontale Teile, die Tiles genannt werden. Für jedes einzelne Tile kann nun definiert werden: Die Länge des Tile, die Startadresse der zugehörigen Bitmap, die Breite der Bitmap, die Bittiefe der Bitmap, Start- und Stop-Bit-Position innerhalb der Bitmap. Man sieht: Es läßt sich auf den Punkt genau festlegen, welche Bildinformation innerhalb eines einzelnen Tile angezeigt werden soll. Zusätzlich kann jedes Tile per Overlay mit einem Rahmen umgeben werden. Dieser Rahmen ist in seiner Farbe für den gesamten Bildschirm einstellbar und kann wahlweise an jeder der vier Kanten einzeln zugeschaltet werden (oben, unten, links und rechts). Bei entsprechend geschickter Programmierung der verketteten Liste kann man ohne große Probleme ein Fenster wie in *Bild 20* darstellen. Die verkettete Liste wird in Echtzeit während des Bildaufbaus aus dem Bildspeicher gelesen. Durch Umprogrammieren der Liste kann der Bildaufbau mit sehr hoher Geschwindigkeit manipuliert werden. Fenster können verschoben werden, ohne ein einziges Bit der eigentlichen Bildinformation zu berühren. Es lassen sich nahezu beliebig viele Fenster mit jeweils unterschiedlicher Bittiefe beliebig auf dem Bildschirm anordnen, wobei jedes Fenster einen Teil einer größeren Bitmap darstellen kann. Panning, horizontales Softscroll und Zooming sind für den GDP 82786 ein Kinderspiel.

Über eine Spezial-Funktion lassen sich sämtliche internen Register des Chips in den Speicher übertragen, so daß sogar Multitasking-Betrieb mit Timesharing möglich ist.

Im nächsten Teil des Artikels wird der GDP 82786 noch detaillierter beschrieben, außerdem wird eine funktionsfähige Schaltung für den mc-68000-Computer vorgestellt. Dort sieht man auch, wie einfach sich der Baustein an asynchrone Bussysteme ankoppeln läßt.

Zitat des Monats

Hierdurch wird die bisweilen immer noch etwas schlechte Performance der 68000-Portierung bestimmt an Wirkung verlieren. Aus einer Pressemeldung

Literatur

[1] Hitachi Microcomputer Data Book: CRTC 6845.
[2] Thomson 'MOS Integrated Circuits': EF 9365, EF 9366.
[3] NEC Microprozessors: µPD 7220.
[4] Texas Instruments: Graphics System Prozessor Product Bulletin TMS 34010.
[5] Hitachi: ACRTC Advanced CRT Controller.
[6] Hitachi: ACRTC Preliminary Application Note.
[7] Hitachi: ACRTC User's Manual, 680-1-31.
[8] Intel: 82786 CHMOS Graphics Prozessor, Nr. 231676-001.
[9] Intel: 82786 Hardware Configuration, AP-270.
[10] Intel: 82786 Architectural Overview, AP-259.

Lexikon

Begriff	Erklärung
Clipping (Rectangle)	Rechteckiger Bereich, der zur Eingrenzung von Zeichenoperationen dient. Meist wird die Berechnung von Punktkoordinaten ausserhalb des Clipping-Rectangle fortgesetzt, lediglich die Zeichenoperationen werden unterdrückt
Direkter Speicherzugriff	DMA = direct memory access. Dieser Begriff wird verwendet, wenn neben der CPU auch ein anderes Gerät (z. B. Festplatten-Controller) auf den Hauptspeicher zugreifen kann
Display Prozessor	Funktionseinheit innerhalb eines Grafikprozessors, die für den Bildaufbau zuständig ist. Sie erzeugt die nötigen Adressen und Synchronsignale
Dual ported RAM	RAM-Bausteine mit zwei – von einander unabhängigen – Bussystemen (Ports). Sie werden normalerweise für asynchronen Datenaustausch in Multiprozessorsystemen oder mit DMA-Kontrollern verwendet. Speziell für Graphikanwendungen gibt es die VRAMs, bei denen der zweite Port ein 256-Bit langes Schieberegister ist
GDP	Graphik-Display-Prozessor
Interleaving memory access	Abwechselndes Zugreifen auf den Speicher: Beim Interleaving können zwei Prozessoren (Grafikprozessor und Hauptprozessor) abwechselnd auf den selben Speicher zugreifen. Dadurch entsteht für beide Prozessoren ein transparenter Speicher, da keiner etwas vom anderen bemerkt. Man spricht in diesem Zusammenhang auch vom „shared memory", d. h. ein geteilter Speicher. Beide Prozessoren bekommen einen Teil vom Speicher, was hier allerdings mehr zeitlich zu verstehen ist, d.h. sie bekommen einen Teil der Zugriffszeit auf den Speicher
Maskierbar	Interrupts können durch verschiedene Bedingungen generiert werden. Besteht die Möglichkeit, die einzelnen Bedingungen wahlweise ein- und auszuschalten, so spricht man von 'Maskierung'. Dabei ist die Maske meistens eine Binärzahl, jedes Bit entspricht einer Bedingung.
Plotprozessor	Zeichenprozessor. Funktionseinheit innerhalb eines Grafikprozessors, die Zeichenoperationen (Linien, Punkte, Kreise usw.) ausführen kann. Wichtigster Befehl ist ein Block-Transfer-Befehl, der zum schnellen, punktweisem Verschieben von Speicherblöcken dient.
VRAMs	Sogenannte Video-RAMs. Dual-ported RAMs, die zum einen wie ein normales DRAM angesteuert werden, zum anderen auch sequentiell über ein 256-Bit langes Schieberegister gelesen werden können.

Ulrich Rohde

Modular programmieren

Ihr Weg zu Modula-2

Modula-2 gilt unter Insidern als die kommende Programmiersprache. Nur wer die Vorteile der modularen Progammierung kennt, wird zu einem Programmierer, der auch umfangreiche Software-Systeme entwerfen kann. Lesen Sie, worauf man bei der Strukturierung achten muß.

Es hat gerauscht in der Leserbriefabteilung. Der Stoßseufzer von Jürgen Heisig in Ausgabe 12, der das GOTO betraf, löste ungeahntes Echo aus. Bravo riefen die einen, endlich zeigt es einmal jemand den verklemmten Puristen, die uns wirklich jedes GOTO verbieten wollen. Und pfui riefen die anderen, wie kann man denn heute noch solche Ansichten äußern. Unsere Reaktion darauf: Wir zeigen in einer Artikelserie, was es mit dem modularen und strukturierten Programmieren auf sich hat und weshalb ein Profi nicht mehr ohne auskommt. Dabei soll nicht verschwiegen werden, daß es heute viele gut ausgetüftelte Basic-Programme gibt, die Großes leisten und dabei heftig GOTOs verwenden. Aber je unübersichtlicher das Gelände wird, das ein Programm bearbeiten soll, je mehr Personen solch ein Programm erarbeiten oder betreuen müssen, desto eher werden Verfahren notwendig, um mit solcher Komplexität nicht nur kraft Genialität fertig zu werden, sondern mit den ehrbaren Mitteln guter Handwerkskunst. Nichts anderes nämlich soll durch modulares und strukturiertes Programmieren erreicht werden.

Die Ausgangslage: Es soll ein Programm geschrieben werden

Wenn es bei dieser Ausgangslage nicht gerade nur ein unbestimmtes Gefühl gibt, daß man doch einmal dieses oder jenes Problem programmieren solle, dann gibt es eine präzise Aufgabenbeschreibung für das zu erarbeitende Programm, ein Pflichtenheft. In diesem Pflichtenheft sollte möglichst exakt festgehalten sein, was das geplante Programm leisten soll. Professionals in der Industrie bedienen sich da bestimmter, meist auch ihren individuellen Bedürfnissen angepaßter Formulierungen um sicherzustellen, daß alle an einem Projekt Beteiligten hinreichend genau erfahren, was das Programm leisten soll. Für die Zwecke hier sei nur angenommen, daß eine Aufgabe so genau beschrieben ist, daß feststeht, wie auf irgendwelche zugelassenen Eingaben die vom Programm zu erarbeitenden Ausgaben aussehen (*Bild 1*). Diese Programmier-Aufgabe soll dann von mehreren Programmierern in Angriff genommen werden – etwa so, wie es bei der Küchenarbeit (bei der allerdings mehrere Köche den Brei sehr verderben können) die Aufgabe gibt, ein bestimmtes, kompliziertes Menü zusammenzustellen. Es müssen dann Teilaufgaben vergeben werden, wie zum Beispiel das Kartoffelschälen, das Herrichten des Nachtisches oder der Vorspeise, das Braten des Fleisches, das Garen der Gemüsebeilagen, und, und, und ... Es entstehen also „Teilpflichtenhefte".

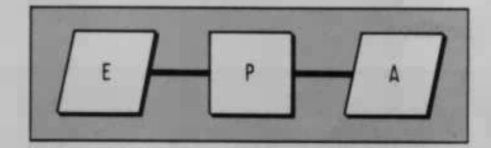


Bild 1. Ein Programm soll auf Eingaben aus einem bestimmten Vorrat eindeutig mit den zugehörigen Ausgaben reagieren

```
10  Kartoffeln schälen
20  Gemüse putzen
30  .....
..........
..........
```

Bild 2. Kochen in der Basic-Küche. Ein Ablauf wird präzise festgehalten

Küche und Computer

Eine Küche und ein Computer haben bei näherer Betrachtung mehr gemeinsam, als man denken würde. Stellen Sie sich zunächst vor, Sie seien allein in der Küche und sie müßten allein kochen. Dann beginnen Sie sicher damit, sich ungefähr einen Ablaufplan zurechtzulegen: Zuerst müssen Kartoffeln geschält werden, dann muß das Gemüse geputzt werden, dann beides in den entsprechenden Topf zum Garen. Das Fleisch danach würzen und ab in die Pfanne, dann .... Es ist nicht absurd, sich vorzustellen, daß man den Plan in Form eines Basic-Programmes niederlegen würde, das in der Reihenfolge der Zeilennummern in jeder Zeile und dort in der Reihenfolge der Nennung der einzelnen Tätigkeiten den ganzen Ablauf präzise darstellt (*Bild 2*).

Es gibt wahre Künstler in der Küche, die nach solchen „geradeaus" geschriebenen Programmen bestens kochen können. Stellen Sie sich nun vor, daß ein Helfer einsetzbar sei, der mitarbeiten könnte, wenn man ihm nur die rechten Anweisungen gäbe. Es kommt nun nochmals der Spruch: mehrere Köche verderben den Brei (an dieser Stelle sei gleich betont, daß in den großen Küchen der Profis sehr wohl mehrere Köche arbeiten und keineswegs den Brei verderben). In der individuellen Küche eines einzelnen Genies gilt dieser Spruch. Und zwar, weil dort keine Methode entwickelt ist, den vorgenannten Ablaufplan, das Kochprogramm, einem Helfer in Teilen so mitzuteilen, daß der ohne Probleme sinnvoll mitarbeiten kann.

Die Zerteilung von Aufgaben ist natürlich

Um Teil-Aufgaben delegieren zu können, muß man die Gesamtaufgabe zerlegen. Ein Koch, der nicht in der Lage ist, sein Tun so zu zergliedern, daß jemand anderes mitkochen kann, wird immer allein kochen müssen – was er vielleicht sogar bewußt oder unbewußt bevorzugt, weil dann auch aller Ruhm für ihn bleibt. Ein Programmierer aus der Basic-Küche, um nun konkret zu werden, besitzt – auch wenn er den Willen hätte, nicht die Mittel, um zu delegieren. Zwar könnte er sagen: bitte, kochen Sie von Zeile 200 bis Zeile 300, ich selbst werde von 100 bis 200 kochen und dann weiter von 300 bis 400, aber es gibt besondere Hindernisse in Basic, die so eine Handlungsweise wenig erfolgsträchtig machen.

Zunächst sei gesagt, daß sich auch jeder Basic-Programmierer vor dem Niederschreiben der ersten Zeile überlegen sollte, wie die Gesamtaufgabe lautet und wie sein Programm in etwa gegliedert sein muß, damit es das tut, was es soll. Im Grunde

muß also jeder Basic-Programmierer ebenso sein Programm in sinnvolle Teile zerlegen, wie es die Puristen empfehlen. Da wird es einen Eingangsteil geben, der die Initialisierungen aller später benötigten Dateien und anderer Resourcen vornimmt. Da wird es vielleicht einen Kommunikationsteil geben, der den Dialog mit dem Benutzer des Programmes abwickelt und einen Bearbeitungsteil, der die eigentliche Bearbeitung der Daten vornimmt. Man kann beim Programmieren in Basic systematisch vorgehen. Es ist aber in Basic besonders schwierig, einen einmal gewonnenen logischen Zusammenhang einzelner Teilaufgaben dem entstehenden Programm aufzuprägen. Das liegt daran, daß Basic fortlaufende Zeilennummern benutzt und der Basic-Prozessor zwar mit GOTO und GOSUB in bestimmte Programmregionen geschickt werden kann, diese aber zu sehr als gleichwertig mit dem übrigen Programm interpretiert. Zum Beispiel würde ein Basic-Prozessor ein mit niedriger Zeilenzahl geschriebenes Unterprogramm, wenn es nicht mit GOTO übersprungen würde oder nicht das Hauptprogramm selbst schon vorher mit END abgeschlossen wäre, einfach als Detail des Hauptprogrammes absolvieren. Dies ist eins der Handicaps, die Basic besitzt, wenn es um das systematische Programmieren geht. Trotzdem könnte man sagen, dagegen hilft ja ein einfacher Sprung, ein GOTO, einmal mehr ein Beweis, daß GOTO nützlich ist und dem ausgiebigen Gebrauch von Unterprogrammen steht nichts mehr im Weg.

Delegieren

Jetzt kommt ein weiterer Grund, weshalb es ein Basic-Koch schwer hat, Hilfsköche zu beschäftigen: In Basic ist die Zuteilung der Töpfe zum Aufbewahren des Kochgutes und die Weitergabe der Kochergebnisse besonders schwierig. Wenn es Ihnen gelungen wäre, jemandem mitzuteilen, er solle in den Zeilen von 1000 bis 2000 ein Unterprogramm für die Kommunikation mit dem Benutzer nach den und den Vorgaben schreiben, dann müßte der Auftragnehmer sich mit Ihnen vor dem Schreiben des Unterprogrammes genau abstimmen, welche Variablennamen er verwenden darf. Denn in Basic gelten alle Variablennamen für das gesamte Programm. Das macht mehr Probleme, als bei anderen Sprachen, weil beim Programmieren einerseits durch Verwendung eines Variablennamens an mehreren Stellen ungewollte Nebeneffekte auftreten können und weil andererseits in Unterprogrammen Variablennamen nicht mehr freizügig gewählt werden können.

Es sei wiederum betont, daß diese Probleme nicht darin bestehen, daß man dieses oder jenes in Basic nicht progammieren könne, sondern darin, daß die Struktur des Programmierten im Progamm sowenig zum Ausdruck kommt, daß nicht nur das Verständnis schwierig ist, sondern auch die Erarbeitung des Programmes selbst, weil Arbeitsteilung nicht unterstützt wird. Das Variablennamen-Problem auf die Küche übertragen hieße, daß es nicht klar ist, wer von den Köchen welchen Topf wann benutzen darf. Und darüberhinaus gibt es in der Küche keinen Stellplatz, an dem die Ergebnisse einer Kochprozedur zur Übergabe an die nächste Kochstufe abgelegt werden können.

Modula ist anders

Die Sprache Modula-2 ist entwickelt worden, um die eben anhand von Basic angedeuteten Schwierigkeiten auszuräumen. Dabei war die Absicht bei der Konstruktion von Modula-2 noch weitreichender: Es sollten in Modula-2 auch parallel laufende Prozesse besonders leicht formuliert werden können und außerdem einige Aufgaben leichter programmiert werden können, denen die Programmiersprache Pascal wegen ihrer strengen und logisch ziemlich unflexibel gestalteten Definition großen Widerstand entgegensetzt. Damit ist gemeint, daß Pascal seinen Benutzern wenig Freiraum in der Wahl der Mittel läßt, wenn es um die Programmierung von systemnahen Angelegenheiten geht. Um klar zu machen, was der Konstrukteur sowohl von Pascal als auch Modula-2, der Schweizer Informatiker Nikolaus Wirth, mit Modula-2 im Sinn hatte und weshalb er das Konzept von Pascal nicht einfach in der Art weiterentwickelt hat, wie es zum Beispiel Borland mit Turbo-Pascal tat, müssen einige Begriffe geklärt werden. Es ist jedenfalls so, daß Modula-2 Pascal nicht ablösen wird, weil es im absoluten Sinn besser ist. Vielmehr ist Modula-2 eine Programmiersprache, die Formalismen eingebaut hat, die dem Programmierer konstruktive Elemente großer Mächtigkeit in die Hand geben. Besonders nützlich sind diese Elemente, wenn es um die Formulierung großer Programmsysteme geht. In diesem Feld hat Pascal als „Lehrsprache" seine Grenzen.

Ein paar Verabredungen

Die folgenden Definitionen sind aus dem Buch „Softwaretechnologie" von Franz Stettner (*Teubner Verlag*) entnommen, das sich durch präzise Formulierungen auszeichnet und in dem sehr klar auch viele Begriffe aus der Informatik geschildert sind, die nicht im strengen Sinn definierbar sind und die deshalb streitbaren Geistern ein unerschöpfliches Reservoir an Munition geben, mit der Praktiker beschossen werden, die es wagen, irgendetwas einfach geradeaus zu programmieren.

Ein Programm ...
...ist (nach „Normen über Informationsverarbeitung", Hrsg. Deutscher Normenausschuß) eine zur Lösung einer Aufgabe vollständige Anweisung zusammen mit allen erforderlichen Vereinbarungen. Eine Anweisung ist eine in einer beliebigen Sprache abgefaßte Arbeitsvorschrift, die im gegebenen Zusammenhang wie auch im Sinne der benutzten Sprache abgeschlossen ist. Eine Vereinbarung ist eine Absprache über in Anweisungen auftretende Sprachelemente.
Ein Programm kann selbst aus Programmen zusammengesetzt sein (*Bild 3*). Wenn diese

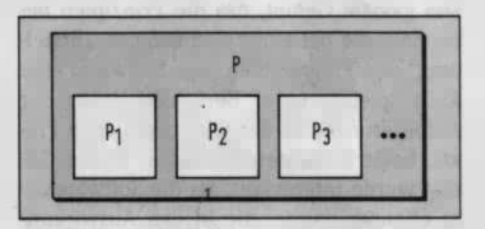


Bild 3. Ein Programm kann aus untergeordneten Programmen zusammengesetzt sein

Teilprogramme eines Programmes gewissen Normen genügen, die die Art der Zusammensetzung betreffen und ihr Schwergewicht in der Bestimmung der Schnittstellen zwischen den einzelnen Teilprogrammen haben, dann spricht man von Moduln. Bei der Andeutung der Nachteile, die Basic besitzt, wenn es um große Programmier-Projekte geht, wurde nur gesagt, daß man keine Mechanismen besitze, die eine Ablage für „fertiggestellte" Menüteile darstellen. Basic-Variablen können zwar Teilprogrammergebnisse überallhin transportieren, aber eben nicht gezielt und kontrolliert anderen Teilprogrammen übergeben. Moduln sollen diese Übergabe an ihren Schnittstellen regeln. Die Konstruktion von Moduln prägt sich also in standardisierenden Einschränkungen aus.

Ein Modul M ist ...
... ein Programm mit den zusätzlichen Eigenschaften:
a) M realisiert eine Funktion,
b) M ist umgebungsunabhängig,
c) M hat nach außen eine standardisierte Schnittstelle,
d) M ist qualitativ und quantitativ handlich, überschaubar und verständlich.

Ein Modul muß also eine Arbeitseinheit sein, nicht nur eine willkürlich zusammengestellte Sammlung von Programmen oder Teilen. Außerdem soll er seine Funktion auch wahren, wenn er in eine andere Umgebung versetzt wird. Mit anderen Worten, ein Programmierer, der die standardisierte Schnittstelle des Moduls kennt, kann ihn einsetzen, ohne sein Inneres analysieren zu müssen.

Modular ...
... heißt ein Programm, das in Moduln zerlegt ist.
Diese Sprachregelung sei an den Beginn gestellt, weil Modula-2 seinen Namen von der Technik bezieht, Programme aus Moduln im oben genannten Sinn zusammenzusetzen. Darüberhinaus steckt vieles, was Pascal auszeichnet, ebenfalls in Modula.

Strukturiert programmieren

Das globale Gebiet, das die Prinzipien behandelt, die bei einer planmäßigen „Erstellung" von Programmen (von Software) eine Rolle spielen, unter Berücksichtigung der Anforderungen in der wirtschaftlichen Praxis, heißt Softwaretechnologie. Dieses Gebiet wurde interessant, als die Softwarekrise offenbar wurde, die in den Auswirkungen der Tatsache bestand, daß Ende der 60iger Jahre die Betriebssysteme der Groß-Computer, die bis dahin aus recht unsystematisch programmierten Teilen bestanden, so groß wurden, daß sie nicht mehr überschaubar waren. Weshalb in diesen Softwaresystemen mehr Fehlfunktionen und Inkompatibilitäten auftraten, als man für vertretbar hielt.
Ein großer Name aus dieser Zeit ist der des Informatikers Dijkstra, der unter anderem damals das GOTO untersucht hatte. Das Motiv war die Feststellung, daß Formulierungen mit GOTO besonders giftig sein können, denn sie schaffen innerhalb von Programmteilen Beziehungen, von denen die betroffenen Programmteile im gewissen Sinn nichts wissen. Um zu verstehen, was da die Schwierigkeit sein könnte, sei nochmals der Begriff Programm betrachtet.
Ein Programm P wird deshalb geschrieben, damit eine bestimmte Funktion erfüllt wird. Eine Menge von Eingaben wird von der Funktion P in eine Menge von Ausgaben abgebildet.

P:E --->A, wobei P(e) = a,
würde ein Mathematiker kurz notieren.

Das Programm ist aus Unterprogrammen und Anweisungen zusammengesetzt, deren Auswahl zum Erreichen eines bestimmten Zieles nicht zwangsläufig ist, sondern – und darin besteht ein guter Teil der Schwierigkeiten – in das Belieben des Programmierers gestellt ist. Das Verdienst Dijkstras ist es, klar ausgesprochen zu haben, daß ein Programm nicht immer der nächstliegenden Idee seines Programmierers folgen sollte, sondern mit gewissen Grundstrukturen aufgebaut sein sollte, die die vorhin genannte Beliebigkeit einschränken und dadurch überschaubar machen.
Betrachtet man die Elemente, die ein Programm aufbauen, dann gibt es Anweisungen, die den Ablauf des Progammes nicht weiter beeinflussen. Gemeint sind zum Beispiel arithmetische Anweisungen, Transportanweisungen usw. Und es gibt Anweisungen, die Programmverzweigungen bewirken, wie IF Bedingung THEN Programmteil 1 ELSE Programmteil 2. Solche Anweisungen nennt man Kontrollanweisungen, weil sie die Programmkontrolle übernehmen. Das GOTO gehört dazu. Die Idee von Dijkstra war nun, beim Programmieren nur wenige universelle Standard-Kontrollanweisungen zuzulassen. Eine Art, diese Einschränkung so vorzunehmen, daß man sicher ist, dennoch alles programmieren zu können besteht in folgender Handlungsanweisung:
Ein Dijkstra-strukturiertes Programm (kurz DS-Programm) entsteht
a) aus Anweisungen die den Kontrollfluß nicht beeinflussen (DS-Anweisungen),
b) aus zwei DS-Programmen DS1 und DS2 durch Verketten zu einer Sequenz DS1DS2,
c) aus Verzweigungsanweisungen der Form

IF B THEN DS1 ELSE DS2

wobei DS1, DS2 und B DS-Programme oder Anweisungen sind,
d) aus Schleifen der Form

WHILE B DO DS1

mit DS-Programmen B und DS1.

Bild 4 zeigt die zugelassenen Konstruktionen in Diagrammform.

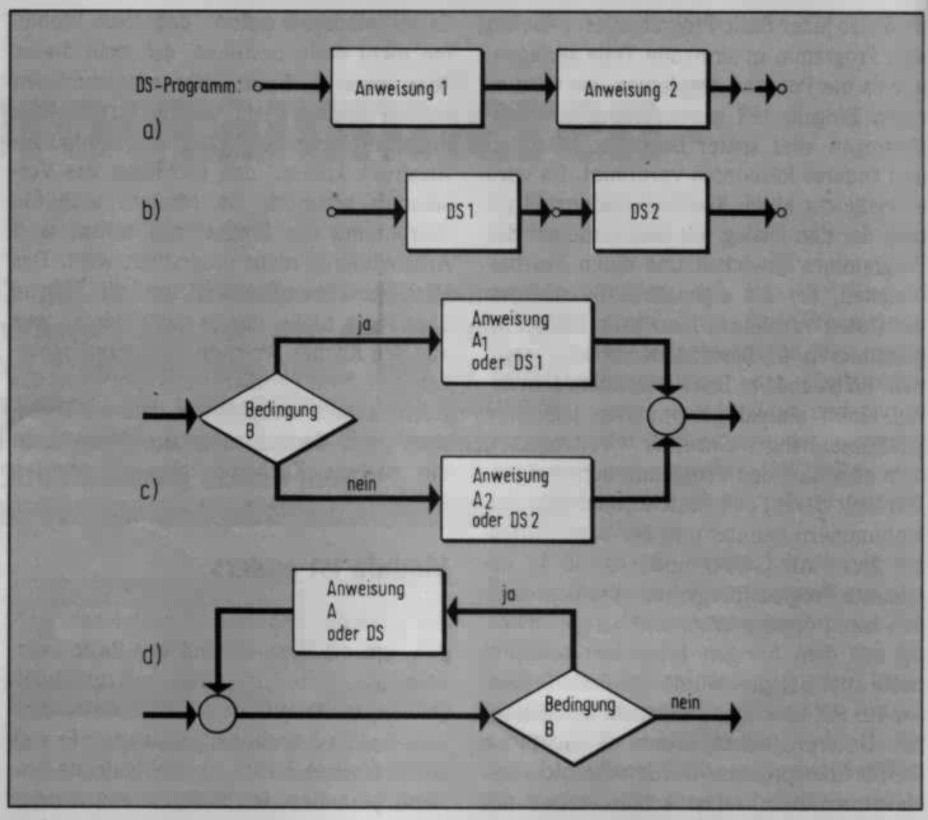


Bild 4. Die Programm-Bausteine für das strukturierte Programmieren

Theorie und Realität

Das oben gezeigte Definitionschema ist ein Baukasten für DS-Programme. Interpretiert man a) so, daß null, eins, zwei, ... , also endlich viele DS-Anweisungen aus einem (hier noch nicht festgelegten) Grundvorrat schon ein DS-Programm bilden, dann kann man sehr viele Programme aus wenigen Grundbausteinen durch Verketten aufbauen. DS-Programme selbst kann man wiederum verketten, wie b) aussagt. Ein DS-Programm B kann die Entscheidung liefern, welches von beiden DS-Programmen DS1

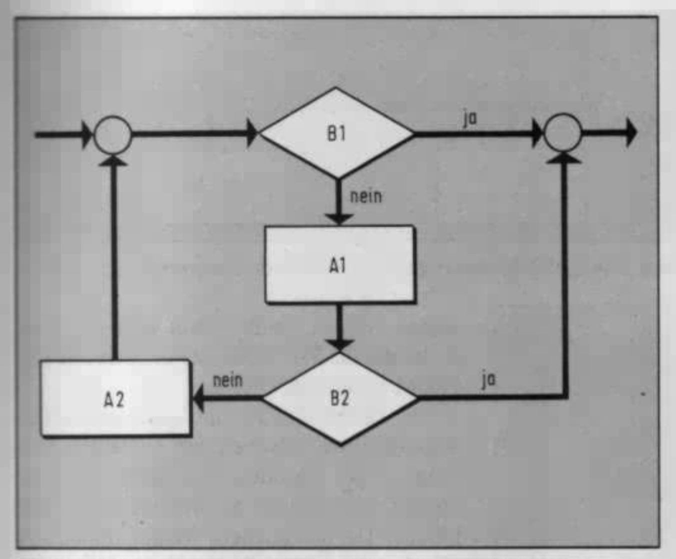


Bild 5. Ein Programm, bei dem aus einer Kontroll-Schleife ausgebrochen wird. Es ist deshalb nicht DS-strukturiert

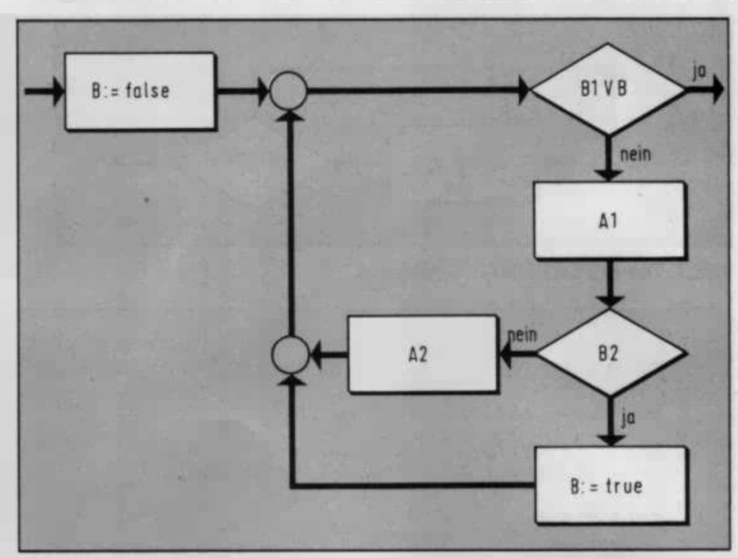


Bild 6. Wenn man eine boolesche Hilfsvariable B einführt, kann man ein Programm schreiben, das funktionsgleich zu dem aus Bild 5 ist

oder DS2 im Rahmen eines größeren DS-Programmes ablaufen soll c) oder wie oft ein DS-Programm ablaufen soll d). Solche Programme haben eine Struktur, denn einzelne Teile sind der Kontrolle anderer unterworfen. Da man DS-Programme mit Kontrollstrukturen wiederum in DS-Programme einbauen kann und dort in übergeordnete Kontrollstrukturen, sind sehr komplexe Programme aus diesen elementaren Strukturen aufbaubar.

In der theoretischen Informatik wird bewiesen, daß man mit DS-Programmen alles programmieren kann, was das Herz begehrt. Um anzudeuten, wie das gehen könnte, zeigt *Bild 5* ein Diagramm, das nicht als DS-Struktur geschrieben werden kann. Wenn der Vorrat der DS-Anweisungen Boolsche Variable und Anweisungen zu deren Behandlung enthält, dann kann man ein DS-Programm entwickeln *(Bild 6)*, das genau das tut, was das Programm aus Bild 5 macht. Das Programm in Bild 6 ist also funktional äquivalent zu dem in Bild 5. Es benutzt einen größeren Befehlsvorrat.

Die an diesen Untersuchungen beteiligten Informatiker waren sehr zufrieden, als klar wurde, daß man mit diesen wenigen Kontrollstrukturen alles programmieren kann. Sie gingen so weit, allen Praktikern die bis dahin viel gebrauchten komplizierteren Kontrollstrukturen abspenstig zu machen und neben den in DS-Programmen enthaltenen Kontrollstrukturen nur noch einige wenige zusätzliche zum Gebrauch freizugeben. Dabei handelt es sich um

REPEAT P UNTIL B,

wobei nach P erst geprüft wird ob P wiederholt wird (*Bild 7*) und die CASE-Struktur (*Bild 8*), mit der Fallunterscheidungen ökonomisch abgewickelt werden können. Die Bilder 4, 7 und 8 zeigen die Strukturen der Kontroll-Anweisungen, die heute in modernen Sprachen eingebaut sind (selbst in vielen Basic-Versionen). Es sind dies die Verzweigung, die bedingte Anweisung, die n-fache Auswahlanweisung (CASE), die Bedingungsschleife (WHILE DO) und die Wiederholungsanweisung (REPEAT UNTIL).

In der Realität hatten sich aber vor allem in Anlehnung an Fortran Praktiken durchgesetzt, die von diesem puristischen Standpunkt stark abwichen. Die Folgen dieser Spaltung der Innung sind offenbar heute noch als Leserbriefseitenfutter zu spüren. In der Fachwelt ist es allerdings heute keine Frage mehr, daß neue Programme nach Möglichkeit strukturiert entworfen werden sollten. Dabei ist unter „strukturiert" heute zu verstehen, daß die Programme sowohl ordentlich hierarchisch in Haupt-Programm (Steuer-Modul) und Unterprogramme (Moduln) aufgeteilt sind, als auch, daß die Kontrollstrukturen möglichst auf die oben aufgezählten Typen beschränkt bleiben. „Möglichst" bedeutet, daß manchmal doch ein GOTO eine Verrenkung in der Struktur wohltuend glätten kann.

Modula-2 bringt's

Jetzt sei noch Niklaus Wirth zitiert, der bei der Entwicklung von Pascal schon alle bisher genannten Aspekte des systematischen Programmierens (und noch einige nicht genannte mehr) berücksichtigt hat:

„Modula-2" entstand aus sorgfältigen Entwurfsüberlegungen als eine Sprache, die alle Möglichkeiten von Pascal enthält, diese jedoch um die wichtigen Konzepte des Moduls und der Multiprogrammierung erweitert. Die wesentlichen Zusätze von Modula-2 bezogen auf Pascal sind:

1. Das Konzept des Moduls und insbesondere die Möglichkeit, ein Modul in einen Definitionsteil und einen Implementationsteil aufzuspalten.

2. Eine systematischere Syntax, die das Erlernen vereinfacht. Insbesondere endet jede mit einem Schlüsselwort beginnende Struktur auch mit einem Schlüsselwort, d. h. sie ist ordentlich geklammert.

3. Das Konzept des Prozesses als Schlüssel zur Multiprogrammierung.

4. Sogenannte niedere, maschinennahe Sprachelemente, mit denen man die strengen Regeln der Konsistenz von Typen aufbrechen kann. Sie erlaubt ferner, Daten einer Modula-2-Struktur auf einen anderen Speicherbereich ohne vorgegebene eigene Struktur abzubilden.

5. Der Prozedurtyp. Er ermöglicht es, eine Zuweisung von Prozeduren an Variable dynamisch vorzunehmen."

Dieses Zitat stammt aus dem Buch „Programmieren in Modula-2" von Niklaus Wirth, das jeder ernsthafte Fan von Modula-2 besitzen sollte. In diesem Buch ist als erstes Beispiel ein Modula-2-Programm aufgeführt, das den größten gemeinsamen Teiler (ggT.) zweier natürlicher Zahlen x und y berechnet. Das Programm beruht auf dem Euklidischen Algorithmus. In Worten geschildert, lautet er etwa so:

Wenn x=y , dann ist x (oder y) der ggT.

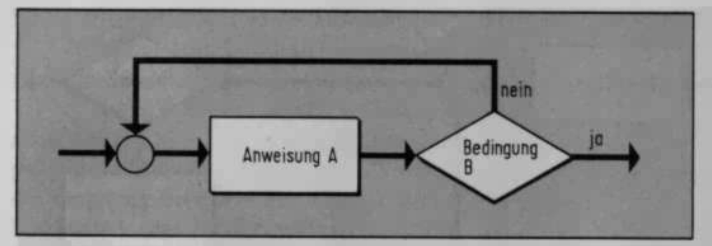


Bild 7. Die REPEAT-UNTIL-Struktur

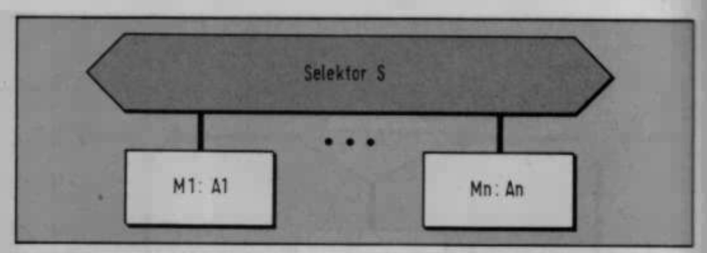


Bild 8. Die CASE-Struktur: genau ein Fall von mehreren

```
MODULE ggT;
  FROM InOut IMPORT ReadCard, WriteString, WriteLn, WriteCard;

  VAR x,y: CARDINAL;
BEGIN
  WriteString("x = ");ReadCard(x);WriteLn;
  WriteString("y = ");ReadCard(y);WriteLn;
  WHILE x # y DO
    IF x > y THEN x:=x-y
             ELSE y:=y-x
    END
  END;
  WriteString("ggT = ");WriteCard(x,6);WriteLn;
END ggT.
```

Bild 9. Der ggT in Modula berechnet

```
Program Euclid;

var x, y, a, b: integer;
begin
  readln (x);readln(y);
  if (x <= 0) or (y < 0)  then  a := 0

                                else
    begin
      a := x; b := y;
      while a <> b do
        if a > b then
               a := a-b
                 else
               b := b-a
    end;
  writeln (a);
end.
```

◀ Bild 10. ggT in (Turbo-)Pascal

```
10 INPUT X,Y
20 IF X=Y THEN GOTO 1000
30 IF X<Y THEN Y=Y-X
40 IF X>Y THEN X=X-Y
50 IF X><Y THEN GOTO 30
1000 PRINT"ggt = ";X
```

Bild 11. ggT in Basic

Wenn x >< y, dann ist der größte gemeinsame Teiler von x und y auch der größte gemeinsame Teiler der kleineren der Zahlen von x und y und der (positiv genommenen) Differenz von x und y. *Bild 9* zeigt das dazu passende Programm in Modula-2. *Bild 10* zeigt den Algorithmus in einer Form programmiert, wie er im Buch Softwaretechnologie enthalten ist. Hier ist allerdings nicht vorausgesetzt, daß x und y aus dem Reservoir der natürlichen Zahlen entnommen sind, sondern aus dem größeren Reservoir der ganzen Zahlen, die ja auch die negativen einschließen. *Bild 11* schließlich zeigt den Algorithmus in Basic. Da geht es nicht ohne GOTO (klassisches Basic vorausgesetzt).

Die Unterschiede

Alle drei Varianten zeigen einige Eigentümlichkeiten der verwendeten Sprache. Beim letzten Programm in Basic sind es die vielen GOTOs, die es etwas unübersichtlich machen, obwohl es sehr kurz ist. Das Pascal-Programm in Bild 10 zeigt, daß es beim Algorithmus auch darauf ankommt, aus welchem Bereich die zu bearbeitenden Daten stammen. Es enthält schon die meisten Elemente, die ein Modula-2-Programm ebenfalls zeigt. Es gibt einen Programmteil vor dem BEGIN, in dem neben dem Namen des Programmes Vereinbarungen über die Typen der verwendeten Variablen stehen. Dabei zeigt sich hier schon im Programm, daß Elemente aus der Menge der ganzen Zahlen verarbeitet werden sollen, was im Basic-Programm nicht so ist, denn dort führen ganze Zahlen, wenn sie negativ sind, zu Katastrophen. Es ist in Basic am Programm nicht leicht zu sehen, daß es bei negativen Zahlen abstürzen wird.

Das Modula-2-Programm in Bild 9 beginnt mit dem Schlüsselwort MODULE und dem Modulnamen, wie ein Pascal-Programm mit dem Schlüsselwort PROGRAM beginnt und seinen Namen nennt. Dann aber folgt ein Zeile, die für Pascal-Kenner ungewohnt ist: FROM InOut IMPORT ...

Solche Zeilen sind Teil der standardisierten Schnittstellen zwischen den einzelnen Moduln eines Modula-Programmes. Hier kommt das Andere an Modula ganz klar heraus. Der aufgelistete Modul importiert offenbar von einem Ding namens InOut die Dinger ReadCard, WriteString, WriteLn und WriteCard. Sie haben davon die richtige Vorstellung, wenn Sie sich denken, daß Modula ein Baukasten ist, der aus von Ihnen und von anderen Programmierern vorgefertigten Baustein-Programmtexten besteht, die jeweils nützliche Bausteine (Prozeduren und Datenobjekte) enthalten. Es gibt in Modula ganze Bibliotheken solcher Texte, vorgefertigte und von Ihnen selbst zu schreibende. Vieles von dem, was Pascal und Basic stillschweigend oder auch per Vereinbarung dem Programmierer zur Verfügung stellen, muß in Modula ausdrücklich für ein Programmiervorhaben aus anderen Moduln, die in Bibliotheken aufbewahrt werden, importiert werden, was dem Vorgang des Bindens in der Assembler-Programmierung entspricht. Damit hat Niklaus Wirth genau die Freiheit in Modula eingebaut, die ein Systemprogrammierer benötigt. Andererseits besitzt man nicht den Komfort anderer Sprachen, in denen viel Service einfach da ist. Aber niemand kann bei der Konstruktion eines Compilers so viel Service in den Compiler und die damit verbundene Sprache einbauen, daß er alle Ansprüche erfüllen könnte, die irgendein Programmierer später einmal an die Sprache stellen möchte. Deshalb ist es ein Durchbruch zur Klarheit, daß Modula-2 nur wenige Kommandos und Befehle zur Verfügung stellt, aber Verfahren, wie man selbst- oder fremdprogrammierte Moduln als Bestandteile der Sprache aufnehmen kann. Daß es im Modula-2-Programm Variablen vom Typ CARDINAL gibt, ist neu für Pascalisten, das sind die Maschinenzahlen, die den natürlichen Zahlen entsprechen. Daß in Modula Groß- und Kleinschreibung eine Rolle spielen und wie die aus dem Modul InOut importierten Prozeduren arbeiten und was InOut für ein Modul ist, wird das nächste Mal genauer beschrieben. □

mc-PC/XT/AT-Referenzliste

Sobald der PC nach eigenen, individuellen Vorstellungen ausgenutzt oder die Kommunikation mit der Außenwelt auf neue, nicht bereits ausgetretene Pfade gelenkt werden soll, kommt der Benutzer nicht daran vorbei, sich mit Funktion und Adressen der BIOS-Routinen, Interrupts und der einzelnen Bausteine zu beschäftigen sowie die Pin-Belegungen der Schnittstellen in Erfahrung zu bringen. Meist läuft das darauf hinaus, entweder in umfangreichen Büchern die nötigen Informationen zu suchen oder aus mehreren, einzelnen Blättern erst zusammenstellen zu müssen.

mc hat in dieser Referenzliste die wichtigsten Daten über Betriebssystem, Bausteine und Schnittstellen für alle PC/XT/AT-Kompatiblen gebündelt, außerdem ist die Pinbelegung des Mikrokanal-Bus für PS/2-Systeme angegeben. Wenn einzelne Modelle Abweichungen aufweisen (zum Beispiel der Siemens PC-D1), so wird darauf hingewiesen. Darüber hinaus ist zu beachten, daß nicht für jeden MS-DOS-Rechner (nur für die absolut IBM-kompatiblen) die aufgeführten Daten gültig sind. So verfügen zum Beispiel der Siemens PC-D, der Philips Yes und der Data General DG/1 über andere Video-RAM-Adressen und andere Schnittstellen-Bausteine. Zusätzliche Funktionen stellen zum Beispiel der Olivetti-M24 (Echtzeituhrbaustein ab 070H), der HP-Vectra (Interrupt-Controller 8259 ab 07CH und ergänzende ROM-BIOS-Aufrufe mit AH = 6FH bei INT 14H und INT 17H) zur Verfügung. Die in der Übersicht aufgeführten ROM-BIOS-Aufrufe dürften für jeden IBM-kompatiblen Rechner zutreffen, unabhängig vom geladenen Betriebssystem (MS-DOS, CP/M, OS/2 usw.).

Die Referenzliste enthält im einzelnen: die ROM-BIOS-Funktionen mit Kurzbeschreibungen der einzelnen Interrupts, die Adressen der Interrupt-Vektoren im Segment 0000, die I/O-Adressen, die Adressen der seriellen Schnittstellen 8250/16450 (nützlich für die schnelle direkte Programmierung), die Adressen des BIOS-RAM-Bereichs im Segment 0040H, die Register des CMOS-RAM und der Uhr, die Aufteilung des Bildschirmspeichers mit verschiedenen Grafik-Karten, die Slot-Belegung und die Belegung des Mikrokanal-Bus für PS/2-Systeme.

ROM-BIOS-FUNKTIONEN

(Alle nicht mit H markierten Zahlenangaben sind dezimal!)

INT 0-7: CPU-Interrupts

```
INT 0:  Division durch Null
INT 1:  Einzelschritt (Trap)
INT 2:  NMI (bei PCs und ATs u.a. Parity-Error)
INT 3:  Breakpoint (Befehl INT)
INT 4:  Arithmetik-Überlauf (Befehl INTO)
INT 5:  80186,80286,80386: Befehl BOUND (vgl. unten!)
INT 6:  80186,80286,80386: Ungültiger Operationscode;
        8088,8086: maskierbarer Interrupt (INTR) mit
        Vektornummer auf dem Bus
INT 7:  80186: ESC-Befehl;
        80286,80386: Coprozessor ist nicht verfügbar
```

INT 5: Print Screen

Adresse der Routine, die beim Drücken der Tastenkombination Shift-PrtSc angesprungen wird (z.B. das ladbare Programm GRAPHICS für eine Grafik-Hardcopy oder, im Textmodus, eine ROM-BIOS-Routine).

INT 08H-0FH: Hardware-IRQ

```
INT 08H: Timer-Hardware-Interrupt (IRQ 0, Systemuhr);
         80186: CPU-interner Timer 0
INT 09H: Tastatur-Interrupt (IRQ 1, Keyboard-Hardware);
         übergibt den Scan-Code einer gedrückten oder
         losgelassenen Taste im I/O-Port 60H. Die Interrupt-
         Routine ergänzt diesen mit dem entsprechenden ASCII-
         Zeichen und schreibt ihn in den Tastaturpuffer
INT 0AH: IRQ 2; AT: kaskadierter 2. Interrupt-Controller;
         PCs: manchmal für COM3 verwendet, sonst frei
INT 0BH: COM2-Interrupt  (IRQ 3, vom BIOS nicht benutzt)
INT 0CH: COM1-Interrupt  (IRQ 4, vom BIOS nicht benutzt)
INT 0DH: LPT2-Interrupt  (IRQ 5, bei PCs evtl. COM4 o.a.)
INT 0EH: Disk-Controller (IRQ 6)
INT 0FH: LPT1-Interrupt  (IRQ 7, vom BIOS nicht benutzt)
```

INT 10H: Bildschirm

```
AH=0: Bildschirm-Modus einstellen. Kein Output; Input:
    AL=0: Text 40x25 monochrom     AL=7:  Monochrom-Karte (Text)
    AL=1: Text 40x25 farbig        AL=8:  manche EGAs: Text 132x25
    AL=2: Text 80x25 monochrom     AL=9-12: reserviert
    AL=3: Text 80x25 farbig        AL=13: EGA-Grafik 320x200 farb.
    AL=4: Grafik 320x200 farbig    AL=14: EGA+Monoc.-Mon.: 640x200
    AL=5: Grafik 320x200 monochr.  AL=15: EGA+Monoc.-Mon.: 640x350
    AL=6: Grafik 640x200 monochr.  AL=16: EGA-Grafik 640x350 farb.
    (Der Modus AL=6 ist beim Schneider-PC1512 farbig, s. INT 15H)

AH=1: Cursor-Größe definieren. Kein Output; Input:
    CH-Bits 0-4: Cursor-Startzeile, CL-Bits 0-4: Cursor-Endzeile

AH=2: Cursor positionieren. Kein Output; Input:
    BH=Bildschirmseite, DH=Zeile, DL=Spalte
```

```
AH=3: Cursor-Position ermitteln
    Input:  BH=Bildschirmseite
    Output: DH=Zeile, DL=Spalte, CX=Cursor-Größe (s.AH=1)

AH=4: Lichtgriffel-Position ermitteln. Kein Input; Output:
    AH=0: Lichtgriffel nicht aktiv; AH=1: Lichtgriffel aktiv
    DH=Zeile, DL=Spalte, CH=Grafikzeile, BX=Grafikspalte (CGA)

AH=5: Bildschirmseite für Textmodi wählen. Kein Output; Input:
    AL=gewünschte Seite

AH=6: Fenster nach oben scrollen. Kein Output; Input:
    AL=Zeilenzahl; CX=links oben; DX=rechts unten; BH=Attribut

AH=7: Fenster nach unten scrollen; siehe AH=6

AH=8: Zeichen mit Attribut an Cursor-Position lesen
    Input:  BH=Bildschirmseite
    Output: AL=Zeichen, AH=Attribut

AH=9: Zeichen mit Attribut an Cursor-Position ausgeben
    Kein Output; Input: CX=Zeichenzahl, AL=Zeichen, AH=Attribut
    (der Cursor wird dabei nicht weiterbewegt)

AH=10 (0AH): Zeichen ohne Attribut an Cursor-Pos.schreiben.
    Kein Output; Input: CX=Zeichenzahl, AL=Zeichen
    (der Cursor wird dabei nicht weiterbewegt)

AH=11 (0BH): Farb-Palette für Grafik setzen. Kein Output; Input:
    BH=Farbpaletten-Nummer (0-127), BL=Farbwert (für 320x200!)

AH=12 (0CH): Grafik-Punkt setzen. Kein Output; Input:
    DX=Zeile, CX=Spalte, AL-Bits 0-6: Farbe, AL-Bit-7=1: XOR,
    bei EGA: BH=Bildschirmseite

AH=13 (0DH): Grafik-Punkt lesen
    Input:  DX=Zeile, CX=Spalte; bei EGA: BH=Bildschirmseite
    Output: AL=Farbe des Punkts

AH=14 (0EH): Zeichen ausgeben, Cursor weiter. Kein Output;
    Input: AL=Zeichen, BL=Farbe (falls Farbmodus gewählt)

AH=15 (0FH): Bildschirm-Modus ermitteln. Kein Input; Output:
    AL=Bildschirm-Modus (siehe AH=0)
    AH=Zahl der Zeichen pro Zeile
    BH=Aktuelle Bildschirmseite

AH=16 (10H): EGA-Farbpaletten-Register setzen
    AL=0: Register BL auf Farbe BH setzen
    AL=1: Overscan-Register auf Farbe BH setzen
    AL=2: Farbpaletten und Overscan mit 17-Byte-Tabelle
          ab ES:DX setzen; Byte 0-15=Paletten, 16=Overscan
    AL=3: BL=0: Intensivierung umschalten; BL=1: Blinken umsch.

AH=17 (11H): EGA-Zeichengenerator-Routine
    AL=0: Zeichengenerator ES:BX in Block BL (0-3) laden;
          Input ferner: BH=Byte je Zeichen, CX=Zeichenzahl,
          DX=Offset des 1. Zeichens im Block
    AL=1: Monochromen 8x14-Zeichensatz aus ROM in Block BL laden
    AL=2: 8x8-Zeichensatz aus ROM in Block BL (0-3) laden
    AL=3: Zeichensatz wählen; Blockwahl mit BL:
          Bit 3=1: Bits 2+3=Block; Bit 3=0: Bit 0+1=Block
    AL=32 (20H): 8x8-Zeichensatz via INT-1FH-Vektor ES:BX laden
    AL=33 (21H): Zeichens.ES:BX aus RAM laden; CX=Byte je Zeichen,
          BL=1: 14 Zeilen, BL=2: 25 Zeilen, BL=3: 43 Zeilen
    AL=34 (22H): ROM-Zeichens. 8x14; BL=Zeilenzahl wie bei AL=33
    AL=35 (23H): ROM-Zeichens. 8x8;  BL=Zeilenzahl wie bei AL=33
    AL=48 (30H): Statusinformation; Input: BH=0:INT 1FH,
          BH=1:INT 44H, BH=2: 8x14 (ROM), BH=3:8x8 (ROM), BH=4:
          obere Hälfte 8x8 (ROM), BH=5: alternativ 8x14 (ROM);
          Output: ES:BP=Zeichengen.-Adr., CX=Byte je Zeichen,
          DL=Zeilenzahl-1

AH=18 (12H): EGA-Konfiguration
    BL=16 (10H): Konfiguration lesen; kein Input. Output:
          BH=0: Farbmodus aktiv, BH=1: Monochrom-Modus
          BL=Speichergröße: 0=64K, 1=128K, 16=192K, 17:256K
    BL=32 (20H): Auswahl der Print-Screen-Routine für 43 Zeilen

AH=19 (13H): Zeichenkette auf EGA ausgeben. Input:
    ES:BP=Startadresse des Strings, BH=Bildschirmseite,
    CX=Zeichenzahl, DX=Bildschirm-Startposition; AL:
    0  Cursor nicht bewegen; BL=Attribut
    1  Cursor weiterbewegen
    2  String enthält abwechselnd Zeichen+Attribute
    3  Wie AL=2, Cursor wird jedoch weiterbewegt
```

INT 11H: Hardware-Test

```
Kein Input; Output: AX-Bits...
0        1: Disk-Laufwerk(e) vorhanden
1        nicht benutzt
2+3      Speicherbänke auf der Hauptplatine
4+5      Default-Video-Modus: 0=illegal, 1=CGA 40x25,
         2=CGA 80x25, 3=Monochrom-Karte 80x25
6+7      Zahl der Disk-Laufwerke, falls Bit 0=1
8        nicht benutzt
9-11     Anzahl der V.24-Schnittstellen
12       1: Game-Adapter vorhanden
13       nicht benutzt
14+15    Anzahl der Parallel-Schnittstellen
```

INT 12H: Speichergröße

```
Kein Input; Output: AX = RAM-Größe in 1-KByte-Blöcken laut
Konfigurations-DIP-Schalter (PC) oder CMOS-RAM (AT)
```

INT 13H: Floppy/Platte

```
AH=0: Disk-System zurücksetzen (keine Parameter)

AH=1: Status der letzten Disk-Operation nach AL:
      0=ok, 1=Ungültiges Controller-Kommando, 2=Adreßmarke nicht
      gefunden, 3=Schreibschutz, 4=Sektor nicht gefunden, 8=DMA-
      Fehler, 9=DMA überlappt 64K-Grenze, 16=CRC-Fehler, 32=
      765-Controller-Fehler, 64=Seek-Fehler, 128=Timeout-Fehler.

AH=2: Sektoren lesen; Input: DL=Laufwerk (0=erstes Disk-Lw.,
      80H=erste Festplatte), DH=Kopf, CH=Spur, CL=Startsektor,
      AL=Sektorenzahl, ES:BX=Pufferadresse. Output: AH=Fehlerstatus
      (s.o.), AL=tatsächlich gelesene Sektoren; C=0: kein Fehler

AH=3: Sektoren schreiben; Parameter wie bei AH=2
```

AH=4: Sektoren verifizieren; Parameter wie AH=2 (ohne ES:BX)

AH=5: Spur formatieren; Parameter wie bei AH=2, jedoch:
ES:BX=Adresse der Formatiertabelle; AL+CL ignoriert

Die folgenden Funktionen existieren meist nur bei ATs und sind hersteller-spezifisch; vom Aufruf aus Anwenderprogrammen wird deshalb abgeraten:

AH=7: Festplatte formatieren
AH=8: Festplatten-Parameter ermitteln
AH=12 (0CH): Spur suchen
AH=13 (0DH): Festplatten-Controller rücksetzen
AH=21 (15H): Disk-Typ lesen
AH=22 (16H): Test, ob Diskette gewechselt wurde
AH=23 (17H): Diskettentyp zum Formatieren setzen

INT 14H: V.24-Schnittstellen

AH=0: Schnittstelle Nr. DX (0=COM1) mit AL-Bits initialisieren:

Baudrate	Bit 7 6 5	Parität Bit 4 3	Stopbits Bit 2	Wortlänge Bit 1 0
110	0 0 0	N x 0	1 0	5 0 0
150	0 0 1	O 0 1	2 1	6 0 1
300	0 1 0	E 1 1		7 1 0
600	0 1 1			8 1 1
1200	1 0 0			
2400	1 0 1			
4800	1 1 0			
9600	1 1 1			

Hinweis: Da INT 14H ohne Puffer arbeitet, ist ein Zeichenempfang bei >110 Bd meist nicht ohne Zeichenverlust möglich.

AH=1: Zeichen in AL senden; Output-Status in AH-Bits:
0 Zeichen wurde empfangen — 5 TX-Holding-Reg.leer
1 Overrun-Fehler — 6 TX-Shift-Reg. leer
2 Paritätsfehler — 7 Timeout-Fehler
4 Break-Signal empfangen

AH=2: Zeichen empfangen; liefert AL=Zeichen, AH=Status (s.o.)

AH=3: Status abfragen; Output: AH=Status wie bei AH=1;
AL=UART-Modem-Status mit folgenden relevanten Bits:
4: CTS 5: DSR 6: RI 7: CD

INT 15H: Sonderfunktionen

a) Kassettenrecorder (nur in den ersten IBM-PCs):
AH=0: Kassettenrecorder-Motor einschalten
AH=1: Kassettenrecorder-Motor ausschalten
AH=2: Datenblöcke von Kassette lesen
AH=3: Datenblöcke auf Kassette schreiben

b) Schneider-/Amstrad-PC1512:
AH=3: Pixelfarbe AL für INT 10H/AH=12 in Register 3DDH schreiben
AH=4: Farb-Maske AL für INT 10H/AH=13 in Register 3DEH schreiben
AH=5: Bildschirm-Randfarbe AL in Register 3DFH schreiben

c) INT-15H-Funktionen in ATs:
AH=80H: Gerät (Device) öffnen
AH=81H: Gerät schließen
AH=82H: Auf Ereignis warten
AH=83H: Joystick abfragen
AH=85H: Test, ob System-Request-Taste gedrückt ist
AH=86H: Bestimmte Zeitdauer warten
AH=87H: Speicherblock-Transfer in/von Extended Memory; CX=Zahl der Worte, ES:SI=GDT-Adresse (Global Descriptor Table). Die GDT umfaßt 6 Deskriptoren mit je 8 Bytes: 2 Byte max. Länge (0=64 KByte), 3 Byte physik. Anfangsadresse, 1 Byte Zugriffsrecht (91H=lesen, 93H=lesen/schreiben), 2 Byte=0 für 80386-Kompatibilität. Die Deskriptoren sind: 1.+2.=reserviert, 3.=Quellbereich, 4.=Zielbereich, 5.+6.=reserviert; Interrupts sind solange verboten!
AH=88H: Liefert Größe des Extended Memory in AX (KByte)
AH=89H: 80286/80386 in den "Protected Mode" umschalten

INT 16H: Tastaturabfrage

AH=0: Auf Tastendruck warten und Zeichen holen. Output:
AL=ASCII-Wert, AH=Scan-Code

AH=1: Prüfen, ob Taste gedrückt. Output:
Z-Flag=1: keine Taste gedrückt
Z-Flag=0: Taste gedrückt; AL=ASCII, AH=Scan-Code

AH=2: Sondertasten abfragen (Shift, CTRL, ALT usw.);
liefert folgende Bits in AL (1=gedrückt/aktiv):
0 Shift rechts — 4 Scroll-Lock aktiv
1 Shift links — 5 Num-Lock aktiv
2 CTRL-Taste — 6 Caps-Lock aktiv
3 ALT-Taste — 7 Insert aktiv

INT 17H: Druckerausgabe

(Hinweis: Beim Siemens-PCD1 wird hiermit eine serielle Schnittstelle angesprochen, die Funktionen werden jedoch emuliert.)

AH=0: Zeichen in AL auf Schnittstelle DX (0=LPT1) ausgeben.
Output = Statusbits in AH (kein Fehler: AH=10H bzw. 90H!):
0 Timeout — 4 Drucker selektiert
1 nicht benutzt — 5 kein Papier
2 nicht benutzt — 6 Acknowledge
3 I/O-Fehler — 7 nicht busy

AH=1: Schnittstelle initialisieren (gewöhnlich überflüssig!);
Output: Status in AH wie bei AH=0

AH=2: Drucker-Status lesen; Output wie bei AH=0. Hinweis: Die Befehlsfolge AND AH,39H, CMP AH,10H, JNZ ERROR springt zur Marke ERROR, wenn der Drucker nicht bereit ist.

INT 18H: ROM-Basic

Beim Original-IBM-PC bzw. -AT wird hiermit das ROM-Basic (Adresse F600:0) gestartet. Bei "Kompatiblen" ist das Basic nicht im ROM, sondern wird vollständig von Diskette bzw. Platte geladen.

INT 19H: Bootstrap

Der Sektor 0 auf Spur 0 von Laufwerk A: oder C:wird ab der Adresse 0000:7C00 geladen und ein Sprung dorthin ausgeführt, um das Betriebssystem von Disk oder Platte zu laden.

INT 1AH: Systemzeit

AH=0: Uhrzeit lesen (die Systemuhr verwendet Timer 0). Output:
CX=Zähler high, DX=low; AL>0: neuer Tag seit letztem Lesen
AH=1: Uhrzeit setzen. Input: CX und DX wie bei AH=0
AH=2: AT: Batterie-gepufferte Echtzeituhr lesen
AH=3: AT: Echtzeituhr setzen
AH=4: AT: Datum von der Echtzeituhr lesen
AH=5: AT: Datum in der Echtzeituhr setzen
AH=6: AT: Alarmzeit in der Echtzeituhr setzen
AH=7: AT: Alarm der Echtzeituhr zurücksetzen

INT 1BH: CTRL-Break-Vektor

Hierher springt das Tastatur-Interrupt-Programm (INT 09H), wenn die Tastenkombination CTRL-Break gedrückt wurde.

INT 1CH: Timer-Interrupt

Wird 18,3mal pro Sekunde angesprungen. Benutzerprogramme, die den Vektor umleiten, sollten am Ende der Interrupt-Routine zur bisherigen Adresse springen (Daisy Chain).

INT 1DH: Bildschirm-Parameter

Adresse der Parameter-Tabelle für INT 10H/AH=0.

INT 1EH: Disk-Parameter

Adresse des aktuellen Disk-Parameter-Blocks.

INT 1FH: Grafik-Zeichensatz

Startadresse des mit GRAFTABL ladbaren 8x8-Grafik-Zeichensatzes für den erweiterten IBM-Zeichensatz (Codes 128-255, CGA+EGA).

INT 70H-77H: AT-IRQ2

INT 70H: Echtzeituhr (IRQ 8)
INT 71H: Ruft INT 0AH auf und simuliert dadurch den IRQ2
INT 72H: COM3-Interrupt bei AT (IRQ 10)
INT 73H: COM4-Interrupt bei AT (IRQ 11)
INT 74H: Reserviert (IRQ 12)
INT 75H: Coprozessor (IRQ 13)
INT 76H: Festplatten-Controller (IRQ 14)
INT 77H: Reserviert (IRQ 15)

VEKTOR-ADRESSEN

Die folgende Übersicht enthält die Adressen der Interrupt-Vektoren im Segment 0000. Jeder Vektor belegt vier Bytes.

Adresse	INT	Bemerkung
0000H	0	Division durch Null
0004H	1	Einzelschritt (Single Step, Trap)
0008H	2	NMI (Non Maskable Interrupt)
000CH	3	Breakpoint (Befehl INT)
0010H	4	Arithmetik-Überlauf (Befehl INTO)
0014H	5	Print-Screen-Routine
0018H	6	AT: Unzulässiger Opcode
001CH	7	AT: Coprozessor nicht verfügbar
0020H	8	IRQ 0; Systemuhr (Timer 0)
0024H	9	IRQ 1; Tastatur-Interrupt
0028H	0AH	IRQ 2; AT: kaskadiert zu 2. 8259
002CH	0BH	IRQ 3; COM2-Interrupt
0030H	0CH	IRQ 4; COM1-Interrupt
0034H	0DH	IRQ 5; LPT2-Interrupt
0038H	0EH	IRQ 6; Disketten-Controller
003CH	0FH	IRQ 7; LPT1-Interrupt
0040H	10H	Bildschirm-Routinen
0044H	11H	System-Konfigurations-Test
0048H	12H	Test auf Speichergröße
004CH	13H	Disketten-/Platten-Routinen
0050H	14H	Serielle Ein-/Ausgabe
0054H	15H	Sonderfunktionen
0058H	16H	Tastaturabfrage
005CH	17H	Drucker-Ausgabe
0060H	18H	Orig.-IBM-PC/-AT: ROM-Basic, sonst IRET
0064H	19H	Boot-Routine
0068H	1AH	Datum und Uhrzeit abfragen/einstellen
006CH	1BH	Break-Vektor
0070H	1CH	Timer-Tick für Anwenderprogramm
0074H	1DH	Video-Parameter
0078H	1EH	Disketten-Parameter
007CH	1FH	Zeiger zu Grafik-Zeichensatz (Code >127)
0080H	20-3FH	MS-DOS-Aufrufe (INT 33H: Maus)
0100H	40H	Diskettentyp-Umschaltung (BIOS)
0104H	41H	Festplatten-Parameter-Tabelle 0
0118H	46H	Festplatten-Parameter-Tabelle 1
0180H	60-6FH	Für Anwenderprogramme frei
01C0H	70H	AT: IRQ 8, Echtzeituhr-Alarm
01C4H	71H	AT: IRQ 9, simuliert PC-IRQ 2
01C8H	72H	AT: IRQ 10, COM3-Interrupt
01CCH	73H	AT: IRQ 11, COM4-Interrupt
01D0H	74H	AT: IRQ 12, frei
01D4H	75H	AT: IRQ 13, Coprozessor
01D8H	76H	AT: IRQ 14, Festplatten-Interrupt
01DCH	77H	AT: IRQ 15
01E0H	78-FFH	Nicht verwendet

I/O-ADRESSEN

000-00FH	DMA-Controller für 8-Bit-Transfers, 8237A
020-021H	Interrupt-Controller für IRQ 0-7/INT 8-0FH; 020H=Kommandoport (z.B. OUT 20H,60H+n = SEOI-Kommando für IRQ n) 021H=IRQ-Freigabemaske (z.B. IN AL,20H; AND AL,11101111B; OUT 20H,AL = IRQ4 erlauben)
040-043H	Timer, 8254; 40H: Timer 0 = Systemuhr, 41H: Timer 1 = RAM-Refresh, 42H: Timer 2 = Lautsprecher, 43H: Steuerreg. für alle Timer
060H	Tastatur-Datenport (Scan-Code), 8042
061H	Systemstatus-Bits (0-3: auch Schreiben mögl.): 0 Timer-2-Gate freigeben 4 Refresh-Tick 1 Lautspr.-Gate freigeben 5 Timer-2-Ausgang 2 RAM-Parity-NMI sperren 6 I/O-Kanal-Err.(AT) 3 I/O-Kanal-NMI sperren 7 RAM-Parity-Err.(AT) Bei PCs bedeuten die Bits 6 und 7 dagegen: 6 Rücksetzen des Tastatur-Strobe 7 Rücksetzen des Tastatur-Interrupt

```
064H          Tastatur-Kommando-Port
066-067H      PC: Konfigurations-Schalter; herstellerabhängig!
070-071H      AT: Echtzeituhr und CMOS-RAM, MC146818
080-087H      DMA-Page-Register-Ports und RAM-Refresh, 74LS612
08EH          AT: Batteriespannungs-Prüfung (Bits 6 und 7)
090H          PS/2: Zentraler Arbiter; Bit 7=System-CPU-Enable,
              6/Write=nur System-CPU erlaubt, 6/Read=NMI aufge-
              treten, 5/Write=600 statt 300 ns Arbiter-Zyklus,
              5/Read=Bus-Timeout aufgetreten, 3-0/Read=Kanalnum-
              mer des gerade aktiven Busteilnehmers
091H          PS/2: Rückmeldung für "Karte selektiert"
092H          PS/2: System-Kontrollport
094H          PS/2: Bit 7=Hauptplatinen-Setup/Enable (0: Setup,
              1: Enable); Bit 5=VGA-Setup/Enable
096H          PS/2: Mikrokanal-Auswahl-Register; Bit 3=Auswahl-
              Freigabe, 0-2=Kanalnummer, 7=Kanal rücksetzen

0A0-0A1H      AT: Slave-Interrupt-Controller f.IRQ9-15, 8259A
0C0-0CFH      AT: DMA-Controller für 16-Bit-Transfers, 8237A
0F0-0FFH      Arithmetik-Coprozessor, 8087/80287

100-101H      PS/2: Karten-ID-Wort; 0=nicht bereit, 1-0FFFH=Bus-
              Master, 5000-5FFFH=DMA-Adapter, 6000-6FFFH=direkte
              Programmkontrolle (z.B.I/O), 7000-8FFFH=Speicher-
              karte, FFFFH=Adapter nicht angesprochen
102H          PS/2: Systemplatinen-I/O-Byte: Bit 0 muß während
              des Zugriffs auf externe Kanäle 0 sein!
103-105H      PS/2: Karten-Optionsauswahl (105H: Bit 7=CHCK)
106-107H      PS/2: Kartenfehlercode lesen, wenn 105H-Bits 6+7=0
              oder Adresszähler auf Karte schreiben

1F0-1F3H      Festplatten-Controller
200-207H      Game-I/O-Adapter (Joystick usw.)
278-27FH      LPT2-Port (vgl. 3BC)
2E8-2EFH      COM4-Baustein, 8250 (PC) oder 16450 (AT)
2F8-2FFH      COM2-Baustein, 8250 (PC) oder 16450 (AT)
300-307H      Prototypen-Karte
378-37FH      LPT1-Port (vgl. 3BC)
380-38FH      SDLC/Bisynch-Karte 2
3A0-3AFH      Bisynch-Karte 1

3B0-3BBH      Monochrom- oder Hercules-Karte (bzw. EGA im
              Monochrom-Modus, die EGA emuliert den 6845):
              3B4 6845-Indexreg.       3B9 Lightpen-FF setzen
              3B5 6845-Datenreg.       3BA Display-Statusport
              3B8 Display-Modus-Port   3BB Lightpen-FF rücks.
3BC-3BEH      LPT-Port auf Monochrom-Karte:
              3BC=Datenbits (0-7); 3BD=Status (Bits 3-7 wie bei
              bei INT 17/AH-Status!); 3BE=Control, Bits:
              0  Strobe                3  Select
              1  Auto Line Feed        4  IRQ zulassen
              2  Drucker-Reset         5-7 nicht benutzt
3BFH          Konfigurations-Schalter auf Hercules-Karte;
              Bit 0: Grafikmodus freigeben (0: nur Textmodus)
              Bit 1: Zweite Speicherseite freigeben, "full mode"

3CEH          EGA-Register-Selektion (nur schreiben); Register:
              00 Set/Reset Reg., Bit 0-3 -> Speicherebene 0-3
              01 Enable Set/Reset Register (Bit 0-3)
              02 Farbvergleichs-Register (Bit 0-3 -> Ebene 0-3)
              03 Data Rotate Register; Bit 0-2=Bitanzahl,
                 Bit 4+3: 01=UND, 10=ODER, 11=EXOR
              04 Speicherebene selektieren (nur Bit 0-2)
              05 Modus-Register; Bit 1+0: 00=alle Ebenen schreiben,
                 01=Ebenen mit Latch laden, 10=Farbe laden;
                 Bit 2: Testmodus; Bit 3: Lesemodus; Bit 4=gerade/
                 ungerade Adresse; Bit 5=IBM-Farbmonitor-Modus
              06 Steuerung der Adressen-Zuordnung im EGA-RAM
              07 Ausblenden einzelner Farben (Bit 0-3)
              08 Bitmasken-Register (für Pixel-Adressierung)
3CFH          EGA-Register-Daten (siehe 03CEH)

3D0-3DBH      Farbgrafik-Karte (CGA) oder EGA, vgl. 3B0-3BBH
3DC-3DFH      Amstrad-PC1512: Register zur Farbsteuerung
3E8-3EFH      COM3-Baustein, 8250 (PC) oder 16450 (AT)
3F0-3F7H      Disketten-Controller, z.B. NEC 765
3F8-3FFH      COM1-Baustein, 8250 (PC) oder 16450 (AT)
```

Hinweis: Die mit "AT" gekennzeichneten Adressen gelten für AT- und PS/2-Systeme. Adressen ohne Vermerk gelten für PCs, ATs und PS/2-Systeme gleichermaßen. (Der Siemens-PCD1 besitzt im Gegensatz dazu eine völlig andere I/O-Adressenbelegung und benutzt auch andere Bausteine.)

8250/16450-PROGRAMMIERUNG

Die folgenden Angaben dienen zur direkten Programmierung der seriellen Schnittstellen, wenn dies aus Geschwindigkeitsgründen erforderlich ist. Alle unten genannten Adressen beziehen sich auf COM1.
Hinweis: Der 8250/16450-Ausgang "OUT2" muß auf log.1 gesetzt werden, wenn Interrupts möglich sein sollen (UND-Verknüpfung mit IRQ).

```
Adresse Register/Inhalt

03F8H   a) Schreiben:   Sendedaten-Register (löscht TxInt)
        b) Lesen:       Empfangsdaten-Reg.  (löscht RxInt)
        c) DLAB=1 (s.u.): Divisor, Low Byte (lesen/schr.)   *)

03F9H   a) Interrupt-Freigabe-Reg.(IER), Bits (Schreiben):
           0  Interrupt bei empfangenem Zeichen (RxInt)
           1  Interrupt, wenn Tx-Hold-Reg.leer (TxInt)
           2  Interrupt bei Framing-/Parity-Error
           3  Interrupt bei CTS-, DSR-, CD- oder RI-Wechsel
        b) DLAB=1 (s.u.):  Divisor, High Byte (lesen/schr.)  *)

03FAH   Interrupt-Identifikations-Reg.(IIR); Bits 3-0 (Lesen):
        001   Kein Interrupt aufgetreten
        000   Modem-Status-Interr.(CTS-,DSR-,CD-,RI-Wechsel)
        010   Transmitter-Holding-Register leer (TxInt)
        100   Zeichen wurde empfangen (RxInt)
        110   Line-Status-Interr.(Framing- oder Parity-Error)

03FBH   Bit 7: Divisor-Latch-Access-Bit; DLAB=1: Schreiben
               der Baudrate in 03F8/03F9H ermöglichen, sonst 0
        Bit 6: Break senden (d.h. TxD=0 für > 1 Zeichen)
        Bit 5: Parity=invertiertes Bit 4 (d.h.konstant 0 bzw.1)
        Bit 4: 1=gerade Parität, 0=ungerade Parität
        Bit 3: 1=Parität senden/empfangen, 0=kein Paritätsbit
        Bit 2: 1=2 Stopbits (bei 5 Bits: 1,5), 0=1 Stopbit
        Bit 1+0: Wortlänge; 00=5, 01=6, 10=7, 11=8 Datenbits
```

```
03FCH   Modem-Control-Register; Bits (Schreiben):
        0  DTR                    3  OUT2 (IRQ-Freigabe)
        1  RTS                    4  Loop-Modus (für Selbsttest)
        2  OUT1 (nicht benutzt) 5-7 nicht benutzt

03FDH   Line-Status-Register; Bits (Lesen):
        0  Data Ready             4  Break empfangen
        1  Overrun Error          5  Tx-Holding-Reg.leer
        2  Parity Error           6  Tx-Schiebereg. leer
        3  Framing Error          7  nicht benutzt

03FEH   Modem-Status-Register; Bits (Lesen):
        0  CTS hat sich geändert          4  CTS-Status
        1  DSR hat sich geändert          5  DSR-Status
        2  RI  hat sich geändert          6  RI-Status
        3  CD  hat sich geändert
```

*) Divisor-Wert (bei Quarzfrequenz 1,8432 MHz) = 115200/Baudrate, also z.B. 48 für 2400 Baud oder 384 für 300 Bd.

BIOS-RAM-BEREICH

Alle folgenden Adressen liegen im BIOS-RAM-Segment 0040H.

Adr.	Bytes	Inhalt
0000H	2	Adresse von COM1 (F8 03 = 03F8H)
0002H	2	Adresse von COM2 (F8 02 = 02F8H)
0004H	2	Adresse von COM3 (meist nicht verwendet)
0006H	2	Adresse von COM4 (meist nicht verwendet)
0008H	2	Adresse von LPT1
000AH	2	Adresse von LPT2
000CH	2	Adresse von LPT3
000EH	2	Adresse von LPT4
0010H	2	DIP-Schalter-Stellung auf Hauptplatine
0012H	1	Flag für Diagnose-Software
0013H	2	RAM-Größe in KByte
0015H	2	Fehlercodes für Diagnose-Software
0017H	3	Tastatur-Status (0017=Shift-Status)
001AH	2	Tastatur-Pufferzeiger 1
001CH	2	Tastatur-Pufferzeiger 2
001EH	32	Tastaturpuffer (15+1 Worte)
003EH	11	Disketten-Controller-Statusbytes
0049H	1	Video-Display-Modus (s. INT 10H, AH=0!)
004AH	2	Zahl der Zeichen pro Zeile
004CH	2	Länge des Video-RAM in Byte
004EH	2	Adresse des Video-RAM
0050H	16	Cursor-Position (je Bildschirm-Seite 2 Byte)
0060H	2	Größe des Cursors
0062H	1	Nummer der aktiven Display-Seite
0063H	2	Adresse des Video-Controllers 6845
0065H	1	Flagbyte für Video-Controller
0066H	1	Farbpalette für Display
0067H	4	Startadresse und Segment eines Zusatz-ROM
006BH	1	Flag für Interrupt
006CH	4	Timer-Count für 8253-Chip (Systemuhr)
0070H	1	Flag für Tages-Wechsel nach 24 Stunden
0071H	1	Flag für CTRL-Break-Taste (Bit 7=1)
0072H	1	Wert=1234H, wenn CTRL-ALT-DEL gedrückt wurde
0074H	4	Festplatten-Controller-Statusbytes
0078H	1	Timeout-Wert für LPT1
0079H	1	Timeout-Wert für LPT2
007AH	1	Timeout-Wert für LPT3
007BH	1	Timeout-Wert für LPT4
007CH	1	Timeout-Wert für COM1
007DH	1	Timeout-Wert für COM2
007EH	1	Timeout-Wert für COM3
007FH	2	Timeout-Wert für COM4
0080H	2	Startadresse des Tastatur-Puffers (=001E)
0082H	2	Endadresse des Tastatur-Puffers (=003E)
0100H	1	AT: 1=Bildschirm-Hardcopy (PrtSc) läuft, 0FFH=Fehler bei PrtSc, sonst 0

Hinweis: Nicht belegte Schnittstellen-Vektoren enthalten 0000.

CMOS-RAM UND UHR

Die folgende Übersicht zeigt die Register des in ATs zu findenden batteriegepufferten Bausteins MC146818. Der Zugriff auf ein Register erfolgt durch Schreiben der Registernummer in den Port 070H und durch anschließendes Schreiben oder Lesen des Ports 071H.

Adresse	Inhalt	Adresse	Inhalt
00H	Sekunden	10H	Disk-Typ A: und B:
01H	Alarm-Sekunden	11H	Reserviert
02H	Minuten	12H	Plattentyp C: und D:
03H	Alarm-Minuten	13H	Reserviert
04H	Stunden	14H	Konfigurationsbyte
05H	Alarm-Stunden	15-16H	RAM-Größe low/high
06H	Wochentag	17-18H	Expansion-RAM-Größe
07H	Tag	19-2DH	Reserviert
08H	Monat	2E-2FH	CMOS-RAM-Prüfsumme
09H	Jahr	30-31H	Extended-RAM-Größe
0A-0DH	Statusreg. A-D	32H	Jahrhundert f.Datum
0EH	Diagnose-Statusbyte	33H	Power-On-Flags
0FH	Abbruch	34-3FH	Reserviert

VIDEO-RAM

	Segment	Offset	max.Seiten
Monochrom-Karte (Textseite 0)	B000H	0-0FFFH	8
CGA-Karte, Textmodus (Seite 0)	B800H	0-0FFFH	4
EGA-Textmodus, Monochr.-Monitor	B000H	0-0FFFH	8
EGA-Textmodus mit EGA-Monitor	B800H	0-0FFFH	8
Hercules-Karte, Grafikseite 0	B000H	0-7FFFH	2
CGA-Karte (Grafik 640x200)	B800H	0-3FFFH	1
EGA-Grafik, Monochr.-Monitor	A000H	0-FFFFH	4
EGA-Grafik mit EGA-Monitor	A000H	0-FFFFH	2
VGA-Grafik mit VGA-Monitor	A000H	0-FFFFH	2

Siemens-PCD1	(Text+Grafik)	F000H	Modus-abhängig
Tandy-2000	(Text+Grafik)	E000H	Modus-abhängig
IBM-PCjunior	(Text+Grafik)	A000H	Modus-abhängig

Im Textmodus belegt jedes sichtbare Zeichen ein Wort, bestehend aus Zeichencode und Attribut-Byte. Letzteres besteht aus folgenden Bits: 0-2=Vordergrund-Farbe, 3=Intensify (hellere Zeichen), 4-6=Hintergrundfarbe. Wenn durch Bit 5 am Mode-Control-Port (3B8H bzw. 3D8H) Blinken erlaubt ist, kann dieses mit Attribut-Bit 7 eingeschaltet werden; sonst steuert Bit 7 die Hintergrund-Helligkeit. Im Monochrom-Modus (MDA, Hercules, EGA) dient die Vordergrundfarbe 01H zum Unterstreichen von Zeichen.

SLOT-BELEGUNG

Hinweis: Pin 1 ist jeweils an der Rückwand-Seite. Die A-Reihe ist von der Bauteilseite aus gesehen rechts, die B-Reihe links. Mit nachfolgendem Minuszeichen markierte Signalleitungen sind invertiert, d.h. aktiv low.

8-Bit-Slots bei PCs und ATs

Reihe B	Nr.	Reihe A
Masse (GND)	1	I/O CHECK-
Reset DRV	2	D7
+5 V	3	D6
IRQ2 (AT:IRQ9!)	4	D5
-5 V	5	D4
DRQ2	6	D3
-12 V	7	D2
Reserviert	8	D1
+12 V	9	D0
Masse	10	I/O CH RDY
MEMW-	11	AEN
MEMR-	12	A19
IOW-	13	A18
IOR-	14	A17
DACK3-	15	A16
DRQ3	16	A15
DACK1-	17	A14
DRQ1	18	A13
DACKQ-	19	A12
CLOCK	20	A11
IRQ7	21	A10
IRQ6	22	A09
IRQ5	23	A08
IRQ4	24	A07
IRQ3	25	A06
DACK2-	26	A05
T/C	27	A04
ALE	28	A03
+5 V	29	A02
OSC	30	A01
Masse	31	A00

AT-kompatible Rechner besitzen bei etwa der Hälfte der Slots je einen zusätzlichen (kürzeren) Slot, der die Datenleitungen D 0-15 sowie die Interrupt-Leitungen IRQ 8-15 herausführt, von den meisten Erweiterungs-Karten jedoch nicht benutzt wird.

Mikrokanal-Bus (PS/2-Systeme)

Rechts: Hauptstecker

Audio-Masse	1	CD Setup-
Audio	2	MADE24
Masse	3	Masse
OSC 14,3 MHz	4	A11
Masse	5	A10
A23	6	A09
A22	7	+5 V
A21	8	A08
Masse	9	A07
A20	10	A06
A19	11	+5 V
A18	12	A05
Masse	13	A04
A17	14	A03
A16	15	+5 V
A15	16	A02
Masse	17	A01
A14	18	A00
A13	19	+12 V
A12	20	ADL-
Masse	21	PREEMPT-
IRQ9-	22	BURST-
IRQ3-	23	-12 V
IRQ4-	24	ARB0
Masse	25	ARB1
IRQ5-	26	ARB2
IRQ6-	27	-12 V
IRQ7-	28	ARB3
Masse	29	ARB/GNT-
Reserviert	30	TC-
Reserviert	31	+5 V
CHCK-	32	S0-
Masse	33	S1-
CMD-	34	M/IO-
CHRDYRTN	35	+12 V
CD SFDBK-	36	CD CHRDY
Masse	37	D0
D1	38	D2
D3	39	+5 V
D4	40	D5
Masse	41	D6
CHRESET	42	D7
Reserviert	43	Masse
Reserviert	44	DS16RTN-
Masse	45	REFRESH-
Stecker-Nase	- 46 -	Stecker-Nase
Stecker-Nase	- 47 -	Stecker-Nase
D8	48	+5 V
D9	49	D10
Masse	50	D11
D12	51	D13
D14	52	+12 V
D15	53	Reserviert
Masse	54	SBHE-
IORQ10-	55	CD DS16-
IORQ11-	56	+5 V
IORQ12-	57	IRQ14-
IORQ12-	58	IRQ15-

Unten: Video-Erweiterungs-Slot (für VGA usw.)

Reihe B		Reihe A
ESYNC	10	VSYNC
Masse	9	HSYNC
P5	8	Blank
P4	7	Masse
P3	6	P6
Masse	5	EDCLK
P2	4	DCLK
P1	3	Masse
P0	2	P7
Masse	1	EVIDEO
Nase ->	-	

Stefan Hänßgen

Minibilder mit Apple-II und FX-80

Mit einer kleinen Routine läßt sich die Hires-Seite des Apple-II in Briefmarkengröße ohne wesentlichen Detailverlust ausdrucken. Die Ausdrucke eignen sich gut zur Diskettenbeschriftung oder als platzsparende „Bildersammlungen". Als Drucker wird der Epson FX-80 verwendet.

Nach Anpassung der Steuerzeichen müßte das Programm auch mit anderen Druckern arbeiten. Nach der Initialisierung des Druckers wird die Grafikseite spaltenweise ausgelesen und 8-Bit-weise an den Drucker gegeben. Dabei wird nur jeweils jede dritte Bildschirmzeile gelesen, also für die ersten 8 Bit die 0., 3., 6. und 9. Zeile (*Bild 1*). Auf diese Weise gibt das Programm eine ganze Zeile aus. Danach findet ein Vorschub (⅓ Punkt) auf dem Drucker statt. Die nächsten 8 Bit werden eine Bildschirmzeile versetzt gelesen, also die 1., 4., 7. und 10. Zeile. Nach einem weiteren Vorschub (⅓ Punkt) und Druck der dritten 8-Bit-Zeile (2., 5., 8. ... Zeile) sind die obersten 24 Zeilen (3 × 8 Bit) fertig ausgedruckt. Der Drucker befördert das Papier um 2⅓ Punkte weiter und der Prozeß wiederholt sich für die nächsten 24 Zeilen, bis das Bild fertig ist. Nähere Erläuterungen sind den Kommentaren im Assembler-Listing zu entnehmen (*Bild 2*).

Dadurch, daß die vierfache Dichte benutzt wird und der Drucker nur ⅓ Punkt Zeilenvorschub macht, wird das Bild gegenüber einem „normalen" Ausdruck extrem verkleinert, wobei die Proportionen erhalten bleiben. Um die Qualität noch zu erhöhen, kann man vorher das Hiresbild von Farben befreien, indem man die kleine Routine „BW" aufruft. Sie wandelt alle vorkommenden Farben in Schwarzweiß-Darstellung um und hellt dadurch das Bild auf.

Besonders bei einem neuen Farbband wäre der Ausdruck insgesamt zu dunkel. Einige Probeausdrucke sind in *Bild 3* zu sehen. □

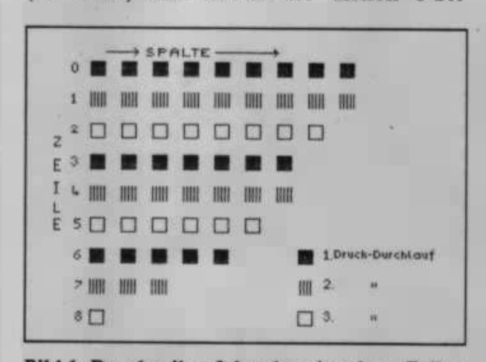
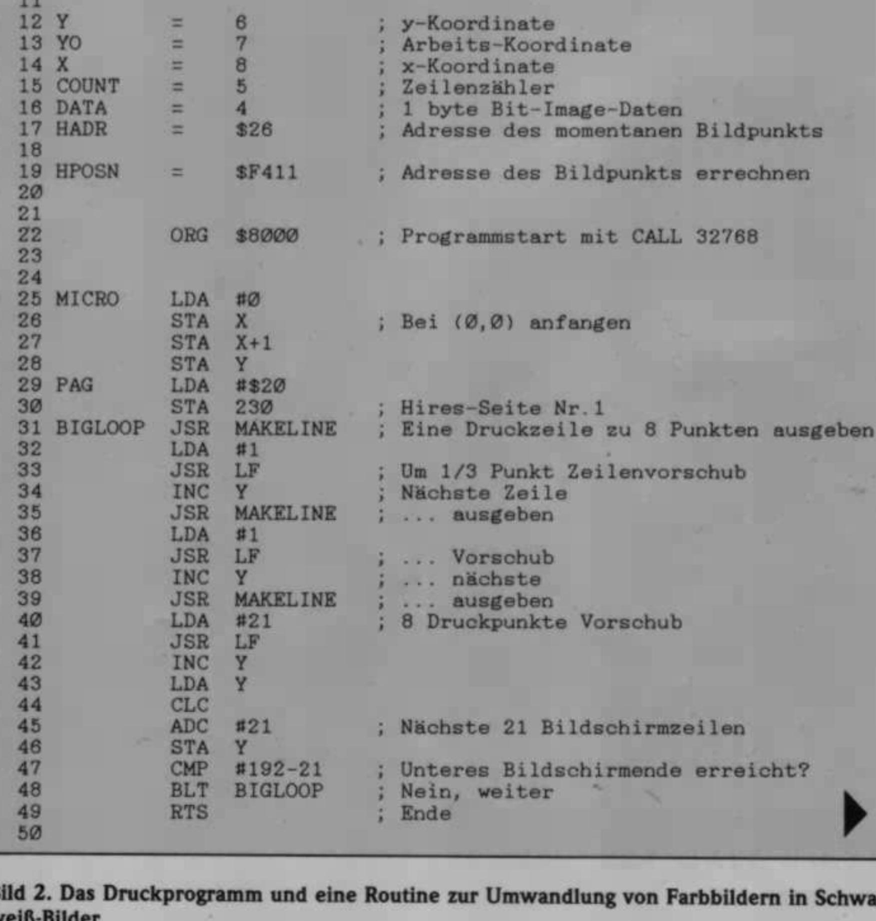


Bild 1. Druckreihenfolge der einzelnen Zeilen



Bild 3. Nette Bildchen kann man mit dem kleinen Programm auf Matrixdruckern anfertigen



```
*****************************************

Y         =    6          ; y-Koordinate
YO        =    7          ; Arbeits-Koordinate
X         =    8          ; x-Koordinate
COUNT     =    5          ; Zeilenzähler
DATA      =    4          ; 1 byte Bit-Image-Daten
HADR      =    $26        ; Adresse des momentanen Bildpunkts

HPOSN     =    $F411      ; Adresse des Bildpunkts errechnen


          ORG  $8000      ; Programmstart mit CALL 32768


MICRO     LDA  #0
          STA  X          ; Bei (0,0) anfangen
          STA  X+1
          STA  Y
PAG       LDA  #$20
          STA  230        ; Hires-Seite Nr.1
BIGLOOP   JSR  MAKELINE   ; Eine Druckzeile zu 8 Punkten ausgeben
          LDA  #1
          JSR  LF         ; Um 1/3 Punkt Zeilenvorschub
          INC  Y          ; Nächste Zeile
          JSR  MAKELINE   ; ... ausgeben
          LDA  #1
          JSR  LF         ; ... Vorschub
          INC  Y          ; ... nächste
          JSR  MAKELINE   ; ... ausgeben
          LDA  #21        ; 8 Druckpunkte Vorschub
          JSR  LF
          INC  Y
          LDA  Y
          CLC
          ADC  #21        ; Nächste 21 Bildschirmzeilen
          STA  Y
          CMP  #192-21    ; Unteres Bildschirmende erreicht?
          BLT  BIGLOOP    ; Nein, weiter
          RTS             ; Ende
```

Bild 2. Das Druckprogramm und eine Routine zur Umwandlung von Farbbildern in Schwarzweiß-Bilder

```
MAKELINE LDA  #155         ; Drucker vorbereiten: <Esc>
         JSR  PRNT
         LDA  #"Z"         ; Bit Image vierfache Dichte <Z>
         JSR  PRNT
         LDA  #24          ; Zahl der zu druckenden Bytes, Low
         JSR  PRNT
         LDA  #1           ; ... und Hi
         JSR  PRNT         ; -> 280 Bytes ausgeben
         JSR  PRLINE       ; Zeile drucken
         RTS
PRINT8   LDA  Y            ; Ein Byte drucken
         STA  YO
         LDA  #8
         STA  COUNT        ; 8 Linien
PRLOOP   JSR  SCRN         ; Bildpunkt lesen
         ROR               ; Farbe (0 oder 1) in Carry schieben
         ROL  DATA         ; Carry ins Datenbyte schieben
         INC  YO           ; 3 Bildpunkte weiter runter
         INC  YO
         INC  YO
         DEC  COUNT        ; Zähler abziehen
         BNE  PRLOOP       ; Noch nicht alle 8 durch
         LDA  DATA
         JSR  PRNT         ; Datenbyte ausgeben
         RTS
PRNT     BIT  $C1C1        ; Drucker aufnahmebereit?
         BMI  PRNT         ; Nein, warten
         STA  $C090        ; Ja, Byte ausgeben
         RTS
PRLINE   JSR  PRINT8       ; Spalte zu 8 Bits ausgeben
         INC  X            ; Nächste Spalte
         BNE  CONT1
         INC  X+1          ; >255
CONT1    LDA  X+1
         BEQ  PRLINE       ; Noch unter 280
         LDA  X
         CMP  #24          ; 280 Bytes ausgegeben?
         BLT  PRLINE       ; Nein, weiter
         RTS               ; Ja, zurück
LF       PHA               ; Accu retten
         LDA  #155         ; <Esc>
         JSR  PRNT
         LDA  #"J"         ; Linefeed in n/216 inch  <J>
         JSR  PRNT
         PLA               ; Accu wieder holen
         JSR  PRNT
         LDA  #$8D         ; Druckkopf an Zeilenanfang
         JSR  PRNT
         LDA  #0           ; X auf Bildschirm-Zeilenanfang
         STA  X
         STA  X+1
         RTS
SCRN     LDX  X            ; X,Y laden
         LDY  X+1
         LDA  YO
         JSR  HPOSN        ; Adresse des Bildpunkts errechnen
         LDA  (HADR),Y     ; Punkt lesen
         AND  $30          ; Bit ausmaskieren
         AND  #%01111111
         BEQ  SCRNO        ; Punkt ist schwarz
SCRYES   LDA  #1           ; Punkt ist weiß
         RTS
SCRNO    LDA  #0
         RTS
```

```
*8000.80BB

8000- A9 00 85 08 85 09 85 06
8008- A9 20 85 E6 20 36 80 A9
8010- 01 20 89 80 E6 06 20 36
8018- 80 A9 01 20 89 80 E6 06
8020- 20 36 80 A9 15 20 89 80
8028- E6 06 A5 06 18 69 15 85
8030- 06 C9 AB 90 D7 60 A9 9B
8038- 20 6C 80 A9 DA 20 6C 80
8040- A9 18 20 6C 80 A9 01 20
8048- 6C 80 20 75 80 60 A5 06
8050- 85 07 A9 08 85 05 20 A4
8058- 80 6A 26 04 E6 07 E6 07
8060- E6 07 C6 05 D0 F0 A5 04
8068- 20 6C 80 60 2C C1 C1 30
8070- FB 8D 90 C0 60 20 4E 80
8078- E6 08 D0 02 E6 09 A5 09
8080- F0 F3 A5 08 C9 18 90 ED
8088- 60 48 A9 9B 20 6C 80 A9
8090- CA 20 6C 80 68 20 6C 80
8098- A9 8D 20 6C 80 A9 00 85
80A0- 08 85 09 60 A6 08 A4 09
80A8- A5 07 20 11 F4 B1 26 25
80B0- 30 29 7F F0 03 A9 01 60
80B8- A9 00 60 00
```

```
**********************************************

ADR      =    6            ; Zeiger auf momentanes Byte
LAST     =    8            ; Bildpunkt, der zuletzt bearbeitet wurde
         ORG  $300         ; Start bei 768
BW       LDA  #$20
         STA  ADR+1        ; Bearbeitung ab $2000 (HGR1)
         LDA  #0
         STA  ADR
         TAY
         STA  LAST         ; Anfangswerte setzen
BWLOOP   ROL  LAST         ; Letzten Bildpunkt in Carry schieben
         LDA  (ADR),Y      ; Byte laden
         AND  #%01111111   ; Hibit ausschalten (gegen Farbeffekte)
         STA  (ADR),Y
         ROL               ; Um 1 verschieben
         ORA  (ADR),Y      ; Übermalen
         STA  (ADR),Y      ; Wieder speichern
         AND  #%10000000   ; Nur Hibit lassen
         STA  LAST         ; ... und als letzten Bildpunkt merken
         INY               ; Nächstes Byte
         BNE  BWLOOP
         INC  ADR+1        ; Page-Grenze, Adresse +256
         LDA  ADR+1
         CMP  #$40         ; Am Ende der Hiresseite ($4000) ?
         BNE  BWLOOP       ; Nein, weiter
         RTS               ; Ja, fertig
```

```
*300.328

0300- A9 20 85 07 A9 00 85 06
0308- A8 85 08 26 08 B1 06 29
0310- 7F 91 06 2A 11 06 91 06
0318- 29 80 85 08 C8 D0 EC E6
0320- 07 A5 07 C9 40 D0 E4 60
0328- 00
```

Klaus Gottschalk

Fig-Forth auf dem mc-68-ECB

Die Anpassung von 68000-Fig-Forth an den mc-68000 wurde in mc bereits beschrieben. Hier folgt die Anpassung an den mc-68-ECB. Dabei stehen die Disketten-Laufwerke durch die beschriebenen Treiber-Routinen für Forth zur Verfügung, Editor und 68000-Assembler lassen sich einbinden.

Hat man sich die 68000-Fig-Forth-Quelle [1] besorgt und mit Hilfe des 68-ECB-Monitors und viel Geduld „hexweise" eingetippt, fehlen nur noch die Schnittstellen-Routinen innerhalb des Speicherauszugs $1D00...$1DFF (*Bild 1*) und deren Einbindung in Forth. Dies geschieht ganz einfach durch Eintragen der Einsprungadresse für die Routinen f_emit, f_key, f_?terminal, f_cr und f_rw (*Bild 2*) in die COLDUSER-Tabelle an den Adressen $36CC...$36D4; also mit dem Monitor-Befehl „S":

```
S36CC 1D40
S36CE 1D44
S36D0 1D48
S36D2 1D4C
S36D4 1D50
```

Von dort aus wird in die eigentlichen Treiber-Routinen emit, key, ?terminal, cr und rw weiterverzweigt. Dann sind – auch mit dem Monitorbefehl „S" – die Parameter für die BUFFER-Verwaltung durch folgende Eingaben festzulegen:

```
S36D6 0080 (B/BUF = 128)
S36D8 0008 (B/SCR =   8)
S36DA 0040 (C/L   =  64).
S36C2 1420 (LIMIT, oder auch =$1840
           bei 2 screens/BUFFER, falls
           gewünscht)
S36B8 0000 (WARNING =0, zunächst, bis
           Disk installiert, dann =1, falls
           Error-Messages gewünscht)
S36C8 8000 (DICTLIMIT hochsetzen, da-
           mit später EDITOR, ASSEM-
           BLER usw. Platz finden)
```

Treiber-Routinen

Die eigentlichen Treiber-Routinen beginnen ab $1D60. Die Beschreibung der Routinen emit, key, cr ist einfach, da hierfür bereits der 68-ECB-Monitor die erforderlichen Funktionen zur Verfügung stellt und daher nur an die entsprechenden Adressen verzweigt zu werden braucht. Etwas umständlicher ist die Funktion ?terminal, da hierbei die BIOS-Funktion „CONST" aufgerufen werden muß. Die genaue Vorgehensweise hierzu findet sich in [4].

Zugriffe auf die Disketten-Laufwerke ermöglicht die Routine „rw". Der Aufruf hierhin erfolgt von Forth aus mit dem Wort R/W, und es müssen vor Aktivierung von R/W die Schreib-/Lese-Flagge, die BLOCK-Nr. und die BUFFER-Adresse auf den Parameterstapel gelegt werden. Die Routine rw holt sich die Parameter in die Register D0, D1 und A0 und errechnet daraus die Laufwerknummer, die Spur und den Sektor auf der Diskette und bewirkt durch Aufruf der entsprechenden BIOS-Funktion den gewünschten Transfer zwischen Diskette und Z80-DMA-Puffer. Außerdem müssen noch die Daten zwischen dem Z80-DMA-Puffer und dem BUFFER innerhalb des 68000-Adreßraums kopiert werden. Der Z80-DMA-Puffer befindet sich innerhalb der Z80-Treiber-Routine ab Adresse 010AH [4]. Die Kommentare im Ausdruck der Assembler-Quelle (*Bild 2*) sollten ausreichen, den genauen Ablauf der Routine zu erläutern.

```
:001D00  4EF8 3662 4E71 4E71  4EF8 369E 0000 0000   ;N.6bNqNqN.6.....
:001D10  0000 0000 0000 0000  0000 0000 0000 0000   ;................
:001D20  0000 0000 0000 0000  0000 0000 0000 0000   ;................
:001D30  0000 0000 0000 0000  0000 0000 0000 0000   ;................
:001D40  6000 001E 6000 0020  6000 0022 6000 002C   ;'...'.. '.."'..,
:001D50  6000 002E 0000 0000  0000 0000 0000 0000   ;'...............
:001D60  4EF9 0020 020C 4EF9  0020 0204 103C 0002   ;N.. ..N.. ...<..
:001D70  4E45 C0BC 0000 00FF  4E75 4EF9 0020 0210   ;NE......NuN.. ..
:001D80  301B 321B 305B 48E7  FFFE 4600 C03C 0001   ;0.2.0[H...F..<..
:001D90  D07C 000D 3601 967C  0500 6400 0008 4243   ;.|..6..|..d...BC
:001DA0  6000 0008 C343 363C  0001 83FC 0020 2401   ;'....C6<..... $.
:001DB0  4842 D47C 0001 2A3C  0000 007F 227C 0050   ;HB.|..*<.."|.P
:001DC0  0214 B03C 000E 6600  0010 1818 32C4 55CD   ;...<..f.....2.U.
:001DD0  FFFA 4E45 6000 000C  4E45 3819 10C4 55CD   ;..NE'...NE8...U.
:001DE0  FFFA 4CDF 7FFF 4E75  0000 0000 0000 0000   ;..L.Nu........
:001DF0  0000 0000 0000 0000  0000 0000 0000 0000   ;................
```

Bild 1. Die Schnittstellen-Routinen

Verwaltung von Diskettenzugriffen

Die Forth-Programme (Quellen) werden im Gegensatz zu CP/M nicht als Dateien gehandhabt, sondern in „screens" gehalten. Dies sind Textbereiche von 16 Zeilen × 64 Zeichen = 1024 Zeichen/screen = 1 KByte. Dieser Bereich kann noch in Blöcke (BLOCK) unterteilt werden. Die Blöcke werden unmittelbar auf die Diskette gelegt; von Spur 0, Sektor 1, an bis zur maximalen Diskettenkapazität. CP/M kann nur Sektoren zu 128 Byte verwalten. Dementsprechend sind die screens in dieser Anpassung auch in Blöcke von je 128 Byte eingeteilt. Es sind also acht Blöcke/screen (B/SCR = 8). Jedem solchen BLOCK entspricht daher auch ein (logischer) Sektor auf der Diskette. Die Blöcke werden von 0 an durchgehend numeriert. Überschreitet die BLOCK-Nr. die Disketten-Kapazität (d. h. eigentlich: BLOCK-Nr. > (Disk-Kapazität × B/SCR)−1), so schaltet die Routine rw auf das zweite Laufwerk weiter.

Innerhalb des Forth-Bereiches werden die screens im BUFFER gehalten. Die Größe dieses Puffers kann (in Einheiten von B/SCR) frei gewählt werden. Dies geschieht durch Festlegen der Konstanten FIRST und LIMIT (Adr. $36C0 und $36C2). Um Zugriffsanforderungen an die Laufwerke zu deren Schonung nach Möglichkeit zu reduzieren, kann LIMIT so weit erhöht werden, daß zwei screens im BUFFER Platz finden. Die Konstante LIMIT ist dementsprechend festzulegen, falls dies erwünscht wird. Die genaue Beschreibung findet sich in [6].

Der mc-68-ECB kommt in einem mc-CP/M-Computer mit „alten" Laufwerken (40 Spu-

ren, einseitig, doppelte Dichte) zum Einsatz, deshalb beziehen sich die Parameter der hier beschriebenen Routine rw auf die Kapazität von 40 Spuren × 32 (log.) Sektoren × 128 Byte = 160 KByte. Die Diskette ist dabei nicht im ECMA-70-Format, sondern durchgehend in doppelter Dichte formatiert, um Zugriff auch auf die erste Spur zu ermöglichen.

Soll ein anderes Format verwendet werden, sind die Parameter in der Assembler-Quelle anzupassen oder die entsprechenden Konstanten an den Adressen

```
$1D98 (= $0500; OFFSET),
$1DAC (= $0020; Sektoren/Spur)
```

zu ändern.

Jetzt kann vom Monitor aus mit „G1D00“ nach Forth (Kaltstart) gegangen werden (zwischendurch Sicherung auf Diskette nicht vergessen!). Der Warmstart erfolgt über „G1D08“.

Die Definition eines kleinen Forth-Wortes „MON“ ermöglicht den Rücksprung von FORTH aus zurück in den Monitor:

```
HEX CREATE MON 4EF9 ( JMP ) ,
20 , 202 , ( $200202 ) SMUDGE
```

Somit kann jederzeit zwischen Monitor und Forth gewechselt werden, um auch die Möglichkeiten des Monitors zu nutzen (z. B. Trace usw.). Soll vom Monitor aus wieder mit Forth weitergearbeitet werden, so denke man daran, daß nur beim *Warmstart* ($1D08) alle bisherige Arbeit erhalten bleibt, während beim Kaltstart ($1D00) eine vollkommene Neuinitalisierung von Forth vorgenommen wird.

Für die Installation des EDITORs [2] ist noch ein Wort „.LINE“ notwendig, das im 68000-Forth-Kern fehlt:

```
: .LINE (LINE) –TRAYLING TYPE ;
```

außerdem ist noch das im EDITOR vorhandene Wort „MATCH“ nach [2] durch das „MATCH“ nach dem Vorschlag in [1] zu ersetzen.

Schließlich kann noch das von *K. Mantei* [1] entfernte Wort „DR1“ falls gewünscht wieder eingeführt werden:

```
DECIMAL : DR1 1280
(= Diskkapazität*B/SCR ) OFFSET ! ;
```

Dieses Wort dient dazu, die Laufwerks-Voreinstellung zu wechseln (vgl. auch DR0). Wie bereits erwähnt, kann ein Zugriff auf das zweite Laufwerk auch dadurch erreicht werden, daß eine screen-Nr. den Bereich des ersten Laufwerks überschreitet. Demzufolge sind z. B. äquivalent

```
DR1  24 LIST mit
DR0 184 LIST (= 160 + 24).
```

Um von Forth aus Ausdrucke auf dem Drukker erzeugen zu können, gibt es hier noch zwei Worte „PRON“ und „PROFF“ (PRinter ON, PRinter OFF), die durch Setzen bzw. Rücksetzen des „prflag“ des 68-ECB-Moni-

```
;       68000 FIG-FORTH Anpassung an mc68kECB
;          K. Gottschalk, Berlin   7.4.87
; ==========================================
spuren:        EQU      40                    ; Spuren/Laufwerk
sek_spur:      EQU      32                    ; logische (!) Sektoren/Spur
; evtl. an eigenes Disk-Format anpassen!
; ------------------------------------------
byte_sek:      EQU      128                   ; bytes/Sektor (log.) n. CP/M
;
diskkapaz:     EQU      spuren*sek_spur*byte_sek/1024
B_BUF:         EQU      byte_sek              ; B/BUF
B_SCR:         EQU      8                     ; B/SCR
z80_dma:       EQU      $010A*2+$500000
OFFSET:        EQU      diskkapaz*B_SCR       ; FORTH OFFSET
; ------------------------------------------
               ORG      $1D00
forthkalt:     JMP      $3662
               NOP
               NOP
forthwarm:     JMP      $369E
; ------------------------------------------
; forth COLDUSER routinenleiste
; in  $36CC bis $36D4 einflicken
               ORG      $1D40
f_emit:        BRA      emit
f_key:         BRA      key
f_?terminal:   BRA      ?terminal
f_cr:          BRA      cr
f_rw:          BRA      rw
; ------------------------------------------
               ORG      $1D60
emit:          JMP      $20020C               ; zum 68K-ECB-Monitor
key:           JMP      $200204
?terminal:     MOVE.B   #2,D0                 ; BIOS-Fkt 2
               TRAP     #5                    ; BIOS 68K ECB
               AND.L    #$FF,D0               ; maskieren
               RTS
cr:            JMP      $200210
;
; forth R/W landet hier und f, blk und buff-addr sind auf Stapel gelegt
rw:            MOVE.W   (A3)+,D0              ; f=0: schreiben
               MOVE.W   (A3)+,D1              ; BLOCK-Nr.
               MOVEA.W  (A3)+,A0              ; BUFFER-Adr.
               MOVEM.L  D0-D7/A0-A6,-(A7)     ; Regs sichern
               NOT.B    D0                    ; lesen=0; schr=1
               AND.B    #$1,D0                ; maskieren
               ADD.W    #13,D0                ; BIOS-Fkt-Nr errechnen
               MOVE.W   D1,D3                 ; blk
IF:            SUB.W    #OFFSET,D3            ; IF (blk>offset)
               BCC      ELSE                  ; in D1 schon richtige blk-Nr
THEN:          CLR.W    D3                    ; DR0
               BRA      ENDIF
ELSE:          EXG      D1,D3                 ; D1:=(blk-offset)
               MOVE.W   #1,D3                 ; DR1
ENDIF:                                        ; D1:=blk , D3:=Lw-Nr
               DIVS     #sek_spur,D1          ; Spur errechnen
               MOVE.L   D1,D2                 ; Sektor
               SWAP     D2
               ADD.W    #1,D2                 ; Sektor errechnet
; ==> D0=BIOS-Fkt ; D1=Spur ; D2=Sektor ; D3=Laufw-Nr ; A0=BUFFER-Adr
; Regs zum DMA-kopieren vorbereiten
               MOVE.L   #B_BUF-1,D5           ; Zaehler 128 mal =Pufferlaenge
               MOVEA.L  #z80_dma,A1           ; A1=Zeiger Z80-DMA
; IF schreiben
               CMP.B    #14,D0                ; BIOS-Fkt 'schreiben' ?
               BNE      lesen
; THEN erst kopieren, dann schreiben
s_schleife:    MOVE.B   (A0)+,D4              ; REPEAT
               MOVE.W   D4,(A1)+              ;   kopiere BUFFER->Z80-DMA
               DBCS     D5,s_schleife         ; UNTIL zaehler=0
               TRAP     #5                    ; schreiben ausfuehren
               BRA      ende
; ELSE lesen, dann kopieren
lesen:         TRAP     #5                    ; lesen ausfuehren
l_schleife:    MOVE.W   (A1)+,D4              ; REPEAT
               MOVE.B   D4,(A0)+              ;   kopiere Z80-DMA->BUFFER
               DBCS     D5,l_schleife         ; UNTIL zaehler=0
; ENDIF
ende:          MOVEM.L  (A7)+,D0-D7/A0-A6     ; Regs restaurieren
               RTS                            ; und fertig R/W
; ==== E N D E ====
```

Bild 2. Fig-Forth-Anpassung an den mc-68-ECB

tors an Adresse $488 die Druckerausgabe zur Konsolenausgabe parallelschaltet bzw. wegschaltet:

```
HEX 488 CONSTANT EPRINT
: PRON 1 EPRINT ! ; : PROFF 0 EPRINT ! ;
```

Zum Schluß noch ein wichtiger Rat, zwischen CP/M-Disketten und Forth-screen-Disketten wohl zu unterscheiden, um bei Diskettenwechsel versehentliches falsches Überschreiben zu vermeiden. Einer Eigenart des mc-CP/M-Computers zufolge können sich nämlich nach einer Schreiboperation z. B. mittels FLUSH noch Daten im BIOS-Puffer befinden, die erst nach einer weiteren Schreib- oder Leseoperation endgültig auf die Diskette geschrieben werden. Wird in diesem Stadium von einer „screen-Diskette" zu einer CP/M-Diskette gewechselt, so merkt CP/M nichts vom dem Wechsel, und nach einem Warmstart (vom Monitor aus mit „E") werden die im BIOS-Puffer verbliebenen Daten auf die falsche Diskette geschrieben! Als Abhilfe wird daher empfohlen, nach „FLUSH" noch einmal den zuletzt bearbeiteten (oder auch irgend einen anderen) screen mit „LIST" aufzulisten, bevor man die „screen-Diskette" entfernt.

Literatur

[1] 68000-fig-FORTH Rel. 1.1. Mountain View Press, Inc., Mountain View, CA 94040.

[2] fig-FORTH Installation Manual. Forth Interest Group, San Carlos, CA 94070.

[3] *Wurdack, S.:* ECB und 68000: mc-68-ECB. mc 1987, Heft 1, Seite 59.

[4] *Wurdack, S.:* Ein Hex-Monitor für den mc-68-ECB. mc 1987, Heft 2, Seite 64.

[5] *Kruppe, U.:* Fig-Forth auf dem mc-68000-Computer. mc 1985, Heft 6, Seite 62.

[6] *Zech, R.:* Die Programmiersprache FORTH. Franzis-Verlag, München.

Turbo-Pascal-Programme optimieren

Kein Software-Paket ist so perfekt, daß man daran nicht noch etwas verbessern oder optimieren könnte. Diese alte Programmiererweisheit trifft auch auf Turbo-Pascal von Borland zu. Daher gibt es auf dem Markt eine ganze Reihe von Programmen und Utilities, die versuchen, die eine oder andere Schwäche von Turbo-Pascal zu beseitigen. Im vorliegenden Fall wurde das amerikanische Programmpaket Turbo-Optimizer von Turbo-Power-Software getestet, das in Deutschland von H&B, Tettnang, vertrieben wird. Voraussetzung für den Einsatz des Turbo-Optimizers bildet Turbo-Pascal mit der Versionsnummer 3.00B, 3.01A oder 3.02A für PC-DOS.

Das Programmpaket besteht aus den drei Teilen Turbo-Library-Compactor (TLC), Turbo-Object-Optimizer (TOPT) und Turbo-Object-Librarian (TOL). Der Turbo-Library-Compactor reduziert die Größe der Laufzeitbibliothek auf das nötige Minimum. Der Turbo-Object-Optimizer versucht, den Programmcode durch nachträgliche Optiminierung effizienter zu machen. Mit dem Turbo-Object-Librarian kann man eine Bibliothek mit vorübersetzten Modulen aufbauen, die bei der Compilierung mit Turbo-Pascal nur noch eingebunden, aber nicht mehr übersetzt werden. Dadurch läßt sich während der Compilierung Zeit sparen.

Der Lieferumfang des Programmpakets besteht aus zwei Disketten (360 KByte) und einem knapp 100seitigen englischen Handbuch. Das Handbuch ist sehr übersichtlich gestaltet und gut verständlich, allerdings in manchen Passagen etwas knapp gehalten. Es ist ratsam, das Handbuch vor dem Arbeiten mit dem Programmpaket einmal durchzulesen, da es wichtige Informationen über die Funktionsweise der einzelnen Module enthält und man ansonsten später böse Überraschungen erleben kann. Die beiden Disketten enthalten insgesamt 93 Dateien, wobei die meisten Dateien zunächst mit dem mitgelieferten Archivierungsprogramm zu expandieren sind. Die drei genannten Programme sind in Turbo-Pascal geschrieben und die zugehörigen Quellen auf der Diskette enthalten. Gute Programmierer können damit den Turbo-Optimizer selbst weiterentwickeln.

Bei Programmen, die nachträglich den Programmcode ändern, kann man sich meist eines gewissen Unbehagens nicht erwehren. Bei einer solchen Nachbearbeitung besteht immer die Gefahr, daß bei dem Versuch, den Code zu optimieren, sich möglicherweise ein Fehler einschleicht. Nicht umsonst werden im Handbuch eine Vielzahl von Fehlermöglichkeiten beschrieben. Die Anwendung des Turbo-Optimizers auf einige meiner Pascal-Programme zeigte allerdings, daß bei Beachtung der Hinweise im Handbuch und einer halbwegs sauberen Programmierung die Optimierung fast immer ohne Fehler abläuft. Über den Nutzen der Optimierung kann man sich aber streiten. So lassen sich beispielsweise Programme mit Overlays und Chain- oder Execute-Aufrufen prinzipiell nicht überarbeiten. Der Turbo-Object-Librarian extrahiert den Code von Funktionen und Prozeduren eines Programms aus einer COM-Datei und speichert ihn in einer Bibliothek. Solche vorübersetzten Module lassen sich dann in einem beliebigen Programm als externe Funktion oder Prozedur vereinbaren. Bei der Übersetzung des Programms bindet der Turbo-Pascal-Compiler dann nur noch die fertig übersetzen Module ein, wodurch sich unter Umständen eine Reduzierung der Übersetzungszeit erreichen läßt. Falls die Module aber von Diskette geladen werden müssen, ist meist der Diskettenzugriff zeitraubender als die Übersetzung der entsprechenden Module durch den Compiler. Da sich die Utilities TLC und TOPT nur auf die vom Compiler erzeugte COM-Datei anwenden lassen, ist der Einsatz des Turbo-Optimizers während der Entwicklungsphase meist viel zu lästig und zeitraubend. Daher dürften die Utilities TLC und der TOPT meist erst zum Abschluß der Entwicklung zum Zuge kommen. Mit der Utility TLC läßt sich praktisch immer die Laufzeitbibliothek um bis zu 10 KByte reduzieren. Eine Option ermöglicht eine weitere Reduzierung des Programmcodes durch Entfernung unbenutzter Funktionen und Prozeduren von eingebundenen Programm-Bibliotheken. Dadurch läßt sich Speicherplatz auf der Diskette sparen, die Ladezeit des Programms vermindern und bei residenten Programmen der Speicherbedarf reduzieren. Mit der Utility TOPT läßt sich meist die Ausführungsgeschwindigkeit des Programms um einige Prozent steigern. Allerdings kann diese Utility auch nicht hexen. Bei einem Programm mit vielen Diskettenzugriffen oder einem schlechten Programmierstil gibt es meist nichts mehr zu optimieren.

Eine Reihe der Funktionen des Turbo-Optimizers sind in Turbo-Pascal mit der Version 4.0 standardmäßig verwirklicht. Wer aber nicht auf die etwas kompliziertere Version 4.0 wechseln möchte, der kann mit dem Turbo-Optimizer sein gewohntes Turbo-Pascal der perfekten Software einen Schritt näher bringen.

Günther Sternberg

Ulrich Heidenreich

Datenmüll schnell beseitigt

Fast alle Textverarbeitungsprogramme legen Sicherungskopien an. Da man oft vergißt, sie nach Fertigstellung des jeweiligen Dokuments zu löschen, wird die Festplatte allmählich mit *.BAK-Dateien gefüllt. Hier hilft eine kleine Utility weiter, mit der man diese Dateien in allen Directories löscht.

Anwender, die auf der Festplatte zahlreiche Unterverzeichnissen haben, werden den Datenmüll erst in einer zeitaufwendigen Sitzung los. Um in ein Unterverzeichnis zu gelangen, muß man den CD-Befehl (Change Directory) eingeben und um die Sicherungskopien zu löschen, muß man den Befehl „DEL ∗.BAK" eintippen.

Der Arbeitsaufwand erleichterte sich wesentlich, wenn der DEL-Befehl Directory-übergreifend arbeiten könnte. Trotz eines Zeitalters, in dem jeder versucht, solche Probleme mit Turbo-Pascal zu lösen, eignet sich dennoch ein GW-Basic-Programm für diese Aufgabe.

Der Shell-Befehl ist sehr leistungsfähig

Für diese Aufgabenstellung ist der Shell-Befehl wie zugeschnitten. Zunächst benötigt der DEL-Befehl Informationen über die vorhandenen Unterverzeichnisse. Diese Informationen liefert der MS-DOS-Befehl TREE. Nun kann der Shell-Befehl zwar beliebige Befehle an den MS-DOS-Kommando-Interpreter übergeben, jedoch Ergebnisse des übergebenen Befehls zurückzuerhalten, scheint explizit nicht möglich sein. So wird meist zu Assembler-Lösungen gegriffen – siehe auch „GW-Basic ruft MS-DOS" [1] – statt MS-DOS entsprechend auszuschöpfen.

Ausgabe-Umleitung von GW-Basic aus

Mit einer Ausgabe-Umleitung des Tree-Befehls in die Datei TREE.$$$ ist diese Rückübermittlung auch ohne Assembler-Klimmzüge realisierbar: SHELL "TREE > TREE.$$$" zwingt die Ausgabe des TREE-Befehls statt auf den Bildschirm in die Datei TREE.$$$. Diese Datei kann dann wie jede andere Textdatei von Basic aus eingelesen und ausgewertet werden. Um die Zugriffszeit so niedrig wie möglich zu halten, siedelt man TREE.$$$ sinnvollerweise auf der RAM-Disk – hier im Laufwerk E – an. Zu beachten ist allerdings, daß je nach MS-DOS-Version der Tree-Befehl eine etwas andere Ausgabe erzeugt.

```
10000 ON ERROR GOTO 10200
10010 PRINT "Lese Unterverzeichnisse ... ":PRINT
10020 OLDDIR$=""
10030 SHELL "TREE>E:TREE.$$$"
10040 OPEN "I",1,"E:TREE.$$$"
10050 WHILE NOT EOF(1)
10060  LINE INPUT #1,X$
10070  SUBDIR=INSTR(1,X$,"C:")
10080  IF SUBDIR<>0 THEN SUBDIR$=MID$(X$,SUBDIR)
10090  WHILE SUBDIR$<>OLDDIR$
10100   OLDDIR$=SUBDIR$
10110   PRINT SUBDIR$;" ";STRING$(35-LEN(SUBDIR$),".");" ";
10120   KILL SUBDIR$+"\*.BAK"
10130   PRINT "*.BAKs gelöscht!"
10140  WEND
10150 WEND
10160 CLOSE
10170 KILL "E:TREE.$$$"
10180 FOR ZEIT=1 TO 7000:NEXT ZEIT
10190 SYSTEM
10200 IF ERL=10120 THEN PRINT "Keine *.BAKs!":RESUME 10140
10210 END
```

Bild 1. Das Programm DELBAK.BAS ist in GW-Basic geschrieben

```
C: ................................. *.BAKs gelöscht!
C:\CAD ............................. Keine *.BAKs!
C:\ENABLE .......................... Keine *.BAKs!
C:\LETTRIX ......................... Keine *.BAKs!
C:\MSDOS ........................... Keine *.BAKs!
C:\MSDOS\TOOLS ..................... Keine *.BAKs!
C:\PM .............................. Keine *.BAKs!
C:\TEXTE ........................... *.BAKs gelöscht!
C:\TURBO ........................... *.BAKs gelöscht!
C:\WIN ............................. Keine *.BAKs!
C:\WIN\PIF ......................... Keine *.BAKs!
```

Bild 2. So sieht die Bildschirmausgabe aus

```
 900    LINE INPUT "Zu löschende Datei/Datei-Gruppe ?";WHAT$

10120   KILL SUBDIR$+"\"+WHAT$
10130   PRINT WHAT$;" gelöscht!"
10200 IF ERL=10120 THEN PRINT "Keine ";WHAT$;"s!":RESUME 10140
```

Bild 3. Das Programm kann auch Dateien mit anderen Dateinamenerweiterungen löschen

```
10070  SUBDIR=INSTR(1,X$,"Suchpfad:")
10080  IF SUBDIR<>0 THEN SUBDIR$=MID$(X$,SUBDIR+10)

10095  IF SUBDIR$="Hauptkatalog" THEN SUBDIR$=""
```

Bild 4. An die deutsche Version von TREE.COM läßt sich das Programm leicht anpassen

Anpassung an verschiedene TREE-Versionen

TREE.COM des kompatiblen NCRDOS setzt so etwa nur das Wörtchen „Suchpfad:" dem Namen des Unterverzeichnisses voran. Andere Versionen des TREE-Programms mögen hier weitere Abweichungen vorweisen; eine Identifikation des Unterverzeichnisses wird aber immer durch kleine Anpassungen möglich sein.

Nach Auswertung der Unterverzeichnis-Namen führt das Programm DELBAK.BAS (*Bild 1*) bei jedem gefundenen Unterverzeichnis den „KILL ∗.BAK"-Befehl durch. Die hierbei eventuell auftretende Fehlermeldung „File not Found" wird mit einer Fehlerbehandlungsroutine durch die Meldung „Keine ∗.BAK" ersetzt (*Bild 2*).

Da manche Anwender dazu neigen, kurze Testprogramme unter dem Namen TEST.COM im gerade aktuellen Inhaltsverzeichnis zu hinterlassen, kann es sich auch als sinnvoll erweisen, das Programm etwas umzuschreiben, um auch solche Dateien zu löschen. Hierzu muß man nur einige Zeilen ändern bzw. hinzufügen (*Bild 3*). In *Bild 4* ist die Anpassung an die deutsche Version von TREE.COM dargestellt.

Literatur

[1] *Feichtinger, Herwig:* GW-Basic ruft MS-DOS. mc 1987, Ausgabe 8, Seite 80.

Jin Jie, Michael Weiss und Georg Schoffa

Simulation eines Digitaloszilloskops

Schnelle 64-Kanal-Datenerfassung über den Druckeradapter des IBM-PC war ein Thema der Ausgabe 1/88. Die Autoren kontrollierten ihr Meßsystem mit einem Assemblerprogramm, das ein Digitaloszilloskop simuliert. Im folgenden Beitrag wird die notwendige Hard- und Software beschrieben.

Zur Darstellung der Meßwerte wurde eine Hercules-Karte verwendet. Sie ermöglicht eine hochauflösende Grafik auf zwei Seiten, von denen jeweils eine auf dem Bildschirm dargestellt wird. Die Karte wird von der GRAPH-X-Software unterstützt [1]. Dabei sind folgende Module von Interesse:

INT10.COM
Der einmalige Aufruf von INT10 implementiert ein speicherresidentes Interface zwischen dem Betriebssystem (BIOS-Interrupt 10H) und der Grafik-Karte.

HGC.COM HGC FULL
Es wird der Bildschirm für die Hercules-Grafik initialisiert

HARDCOPY.COM, PRINTER.DEF
Sie führt die Grafik-Initialisierung des Druckers durch, so daß eine Hardcopy vom Bildschirm zum Drucker gesendet werden kann.

Die Hercules-Grafik teilt den Bildschirm in 720 × 348 Bildpunkte auf. Bei der grafischen Darstellung müssen alle physikalischen Koordinaten in diese Bildschirmkoordinaten umgerechnet werden.

Alle Grafikanwendungen des hier beschriebenen Programmpakets sind in Assembler programmiert, wobei die in INT10 angebotenen elementaren Funktionen verwendet werden. Außerdem werden weitere Grafikfunktionen benutzt, die in den Programmpaketen HERK_G1.ASM und ADC_MAC1.ASM enthalten sind.

Das Programm ADC_OSZI

Um das Meßgerät optimal justieren oder eventuelle Störungen untersuchen zu können, stellt das Programm ADC_OSZI jeweils einen Kanal auf dem gesamten Bildschirm grafisch dar. Das Meßgerät muß hierzu in der Betriebsart 2 arbeiten, d. h. ohne die Triggersignale vom PC zu erwarten. Das Flußdiagramm des Programms ist in *Bild 1*, das ausführlich kommentierte Assembler-Quellprogramm in *Bild 2* zu sehen.

Das die Meßwerte darstellende Bild wird in zwei Phasen aufgebaut. Zunächst wird das alte Bild gelöscht. Danach werden die Koordinatenachsen (Ordinate: Spannung; Abszisse: Zeit) gezeichnet und die Fußzeile für Anweisungen geschrieben. Danach wartet das Programm auf das Synchronisationssignal (START) des Meßgeräts. Nach der Registrierung dieses Signals wird die zeitliche Änderung des Meßwertes on-line gemessen und punktweise grafisch dargestellt. Dazu sind die jeweiligen Spannungswerte in Bildschirm-Y-Koordinaten umzurechnen. Die zeitliche Auflösung ist durch die 720 Abszissenpunkte des Bildschirms gegeben. Die Dauer einer Meßreihe (720 Einzelmessungen) wird durch die Angabe „Zeitachse“ des Menüs „Oszilloskop“ eingestellt.

Das Programm arbeitet normalerweise in der Art eines Speicheroszilloskops, d. h. eine der beiden Grafikseiten wird gezeichnet, während die andere Seite dargestellt wird. Diese Methode ist zur Darstellung niederfrequenter Meßsignale notwendig.

Das Oszilloskop-Programm bietet folgende in der Fußzeile gelisteten Tastaturanweisungen interaktiv an:

„A“ Akkumulieren
Alle Messkurven werden in ein Diagramm eingezeichnet.

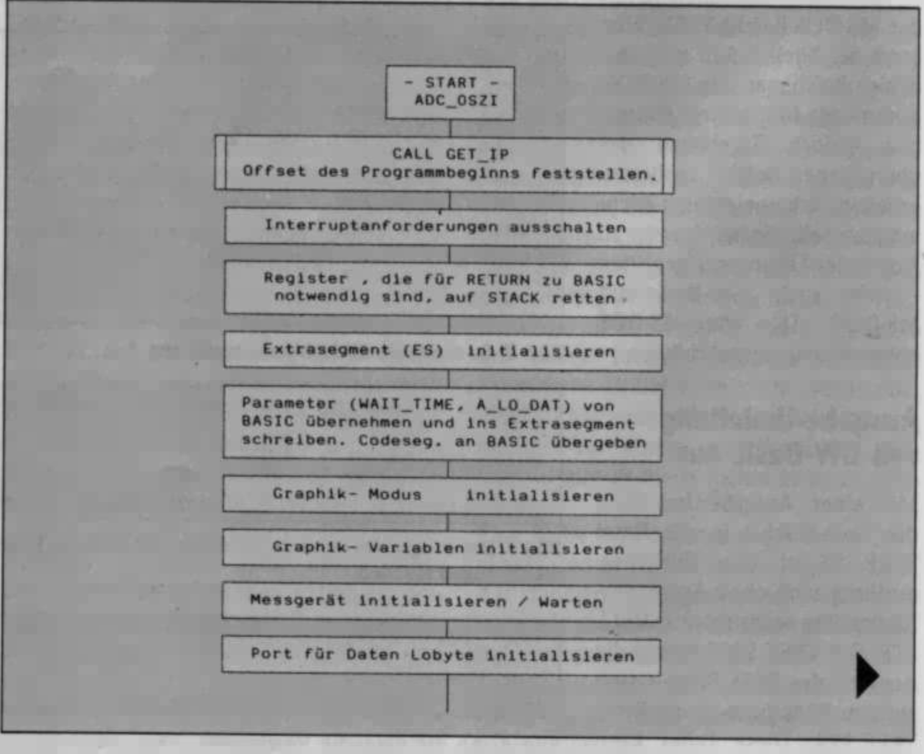
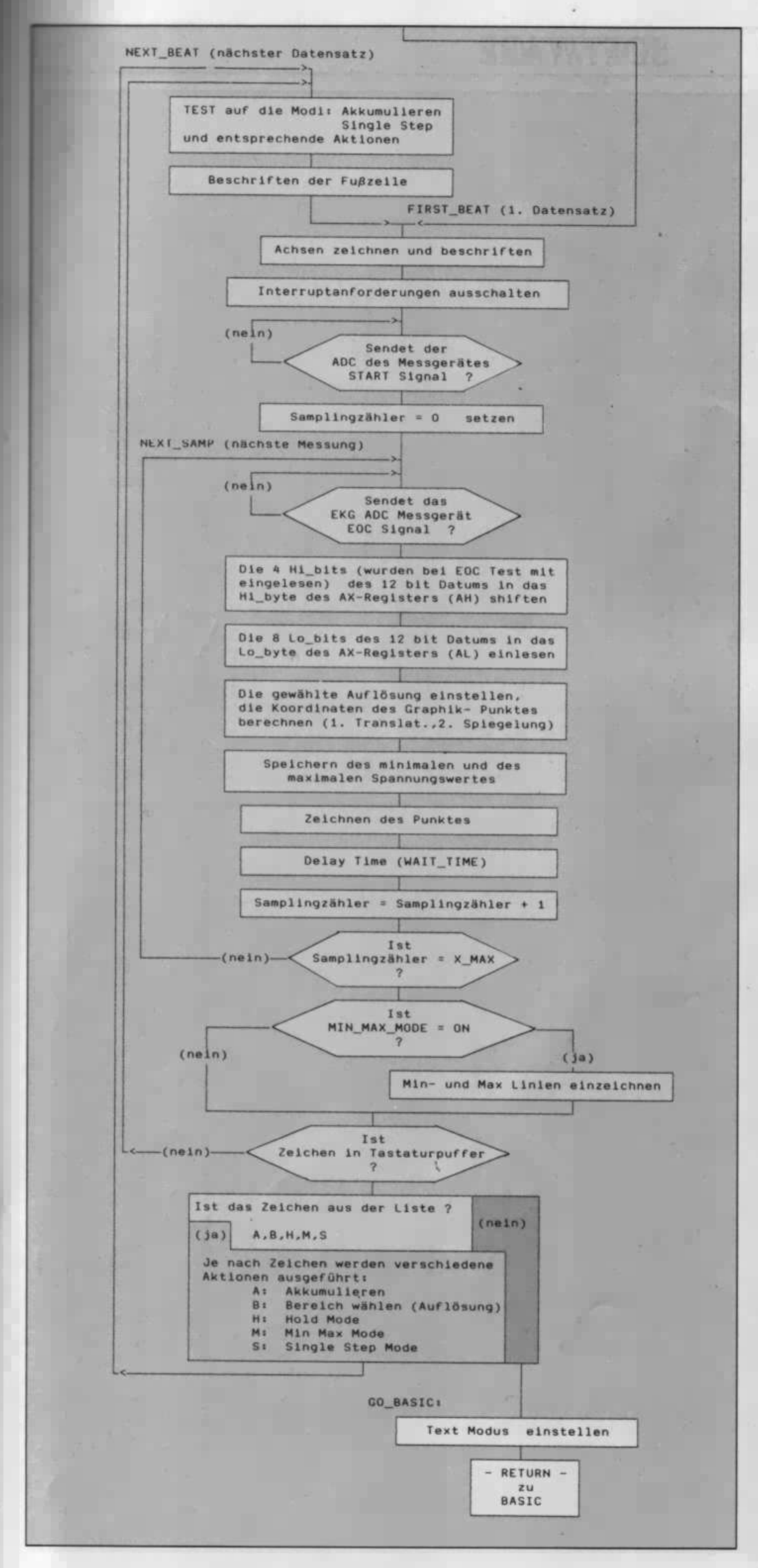


Bild 1. Flußdiagramm zum Assembler-Programm ADC_OSZI



„B" Bereich
Die Auflösung des Bildes kann zwischen 9 und 12 Bit gewählt werden.

„H" Halten
Das Bild bleibt stehen

„M" Min/Max
Der maximale und der minimale Spannungswert werden durch eine Gerade markiert.

„S" Single Step
Der Bildaufbau ist in Echtzeit im Einzelschritt zu sehen.

Alle diese Funktionen werden durch Betätigen der genannten Tasten ein- bzw. ausgeschaltet. Jede sonstige Taste führt zur Rückkehr ins aufrufende Basic-Programm.

Die zusätzlichen Macro-Dateien

Das Programm ADC_OSZI benötigt neben den Routinen aus dem GRAPH-X noch weitere Makro-Routinen. In der Datei HERK-_G1.ASM, die in *Bild 3* aufgelistet ist, sind eine Reihe komplexer Grafikfunktionen enthalten. Sie sind auf der Basis der GRAPH-X Assembler-Anweisungen als Macros definiert. Diese Macro-Bibliothek muß bei der Übersetzung der Grafikprogramme durch den MASM eingebunden werden.
ADC_MAC1.ASM ist eine Macro-Bibliothek, die bei der Übersetzung aller ADC_ASM-Assemblerprogramme durch den MASM eingebunden werden muß, da sie allgemeine Macro-Definitionen enthält. Die Funktion aller dieser Assembler-Programme ist in der Auflistung ausführlich kommentiert. Die Programme sind auch auf einer Diskette erhältlich [2].
Die Schnittstelle zu den Steuerprogrammen in Basic und die mit dem ADC entstehenden Probleme wurden in mc Heft 1/88 beschrieben und geklärt.

Literatur

[1] GRAPH X, Hercules Computer Technology, 2550 Ninth Street, Suite 210, Berkeley CA 94710.

[2] Vollständige Programme als Quellen, Flußdiagramme und ausführlicher Text (64-Kanal-Datenerfassung und Digital-Oszilloskop) erhältlich bei AI-Consult, Fremersbergweg 7, 7505 Ettlingen. Preis 40 Mark (2 Disketten, inkl. Versand).

```
TITLE ADC_OSZI.ASM (DD,DD,WAIT_TIME%,DRUCK_ADAPT%/CSEG%)  Oszilloskop

%OUT Autor M. Weiss      Dez 86
PAGE ,132
COMMENT $ :
-- ADC_OSZI  stellt einen der 64 Kanäle des Messgeräts oszillographisch
dar.
-- ADC_OSZI stellt folgende interaktiv wählbare Modi zur Verfügung:
HOLD, SINGLE_STEP, AKKUMULIEREN, Y_HOLD, RANGE.
-- ADC_OSZI kann an beiden Druckeradaptern betrieben werden.
-- EXE Programm muß in BIN Programm umgewandelt sein. ADC_OSZI.BIN kann
von BASIC geladen und ausgeführt werden.
-- vor dem RUN muß HGC FULL (Bildschirminitialisierung für hochauflösende
Graphik) und INT10.COM (Änderung des BIOS Vektors) aufgerufen worden sein.
-- ADC  Modus:  RANDOM, STROBE d.h.  keine Steuersignale, nur START und
EOC Abfragen.

ALPHABET.ASM muß included werden
HERK_G1 .ASM muß included werden
ADC_MAC1.ASM muß included werden

Adressen:
A_LO_DAT=DRUCK_ADAPT :=Basisadresse (wird vom BASIC Programm übergeben)
        A_LO_DAT                              = 3BCH/378H
        A_HI_DAT                              = 3BDH/379H
        A_STAT_EOC        = A_HI_DAT          = 3BDH/379H
        A_STAT_START      = A_COMMAND         = 3BEH/37AH

spezielle Verwendung von Registern
        AX:     Daten
        AL:     Daten, Status, Befehle, Herkules
        AH:     Daten, Herkules
        BX:     Basisregister
        CX:     Loops
        DX:     Adressen
        DI:     SAMP, Herkules Übergaberegister (X-Koord)
        ES:     Variablen, Parameter
        SS:     STAPEL
        SP:     zur Parameterübergabe
        BP:     um Parameter auf Stack zu erreichen
                 Herkules Übergaberegister (Y-Koord)
$
        IF1
        %OUT 1. Phase begonnen
        ENDIF
        IF2
        %OUT 2. Phase begonnen
        ENDIF

        INCLUDE         ALPHABET.ASM
        %OUT ALPHABET.ASM ist included

        INCLUDE         HERK_G1.ASM
        %OUT HERK_G1 .ASM ist included

        INCLUDE         ADC_MAC1.ASM
        %OUT ADC_MAC1.ASM ist included

;************* VEREINBARUNGEN *****************************************

                ;DATEN
INIT_DAT                EQU     0FFH
                ;ZEITEN ein LOOP_WAIT dauert 20 TZ dh 4,2µs +4 TZ
TIME_DISP               EQU     180
TIME_MULT               EQU     3
                ;BILDSCHIRM
```

Bild 2. Das Assembler-Programm ADC_OSZI simuliert ein Digital-Oszilloskop

```
GROUND                  EQU     (Y_MAX+1)/2
;Eichung der Scala:     (2^10)/10=102,4 Pkt/Volt
SCAL_9                  EQU     (SCAL_10+1)/2
SCAL_10                 EQU     25
SCAL_11                 EQU     SCAL_10*2
SCAL_12                 EQU     SCAL_11*2
G_VOLT_9                EQU     256
G_VOLT_10               EQU     2*G_VOLT_9
G_VOLT_11               EQU     2*G_VOLT_10
G_VOLT_12               EQU     2*G_VOLT_11
K_TRANS_9               EQU     G_VOLT_9 -GROUND
K_TRANS_10              EQU     G_VOLT_10 -GROUND
K_TRANS_11              EQU     G_VOLT_11 -GROUND
K_TRANS_12              EQU     G_VOLT_12 -GROUND
KORR_9                  EQU     K_TRANS_9 +2*GROUND
KORR_10                 EQU     K_TRANS_10 +2*GROUND
KORR_11                 EQU     K_TRANS_11 +2*GROUND
KORR_12                 EQU     K_TRANS_12 +2*GROUND
A_9BIT                  EQU     12-9
A_10BIT                 EQU     12-10
A_11BIT                 EQU     12-11
A_12BIT                 EQU     0

;*************** MACRO VEREINBARUNGEN ******************************

CHANGE_RANGE    MACRO   NEW_RANGE,NEW_KORR,NEW_SCAL
        MOV     RANGE,NEW_RANGE
        MOV     KORR,NEW_KORR
        MOV     SCAL,NEW_SCAL
ENDM

COMMENT $ :     Verwendete Macroaufrufe aus HERK_G1.ASM und ADC_MAC1.ASM

GET_PARAM       P_NAME,V_NAME
PUT_PARAM       P_NAME,V_NAME (BX)
INIT_SEGM       OLD_SEG (AX),DELAY,NEW_SEG (AX)

XCHG_VARIAB     REG,VARIAB_1(REG),VARIAB_2
DIV_2           REG,DIVISOR
BIOS_KEY        K_FKT (AH)

GRAPH_X         G_FKT (AH)
STRING_X        SPALTE(),ZEILE(),<C1,C2,.....,Cn>
PRINT_4_NUMBERS SPALTE(),ZEILE(),NUM(AX),STELLEN(B_XYZ),BASE()

DRAW_X_LINE     Y_POS (BP)
DRAW_AXES       X_POS (DI) ,Y_POS (BP)
SCAL_Y_AXE      DISTANCE,ZERO,<E,I,N,H,E,I,T>,SCAL_MODE
SCAL_X_AXE      DISTANCE,ZERO,<E,I,N,H,E,I,T>,SCAL_MODE

INIT_PAGE
XCHG_PAGE       REG,ONE_PAGE(REG),OTHER_PAGE
DISP_PAGE       PAGE_? (AL)
GRAPH_PAGE      PAGE_?
$

;************STAPEL und DATENSEGMENTE ******************************

STAPEL  SEGMENT WORD STACK 'STACK'

PARAM   STRUC                           ;Parameterliste
SAVE_BP         DW      ?               ;BASIC's BP Register
RET_OFF         DW      ?               ;OFFSET für RET
RET_SEG         DW      ?               ;SEGMENT für RET
P_CSEG          DW      ?               ;Offset von Param. A_LO_DAT (DRUCK_ADAPT)
                                        ;Offset von Rückgabeparameter CSEG
P_WAIT_TIME     DW      ?               ;Offset von Parameter WAIT_TIME
                DW      ?               ;Offset von Parameter CHAN
```

```
P_SAMP          DW      ?       ;Offset von Parameter SAMP
PARAM   ENDS
                DW      20 DUP(?)
STAPEL  ENDS

P_SIZE  EQU     OFFSET P_SAMP-OFFSET RET_SEG    ;Variable f. RET

VARIABLEN       SEGMENT WORD 'DATA'
;ADRESSEN
        A_LO_DAT        DW      ?       ;Basisadresse
        A_HI_DAT        DW      ?
        A_STAT_EOC      DW      ?       ; =A_HI_DAT
        A_STAT_START    DW      ?

        WAIT_TIME       DW      ?       ;Verzögerungszeit zwischen den SAMP
        SCAL_TIME       DW      ?       ;Zeit/Div
        SCAL            DW      ?
        KORR            DW      ?
        MAX_POINT       DW      ?
        MIN_POINT       DW      ?
;MODI
        MIN_MAX_MODE_A  DB      ?
        MIN_MAX_MODE_B  DB      ?
        SIN_STP_MODE_A  DB      ?
        SIN_STP_MODE_B  DB      ?
        AKKUMUL_MODE_A  DB      ?
        AKKUMUL_MODE_B  DB      ?

        RANGE           DB      ?
        ONE_PAGE        DB      ?       ;für HERK Graphik
        OTHER_PAGE      DB      ?       ;für HERK Graphik
        B_SCALE         DB      ?       ;für SCALE_AXES (HERK_G1)
        B_XYZ           DB      ?       ;Byte-Variable für temporäre Zwecke
VARIABLEN       ENDS

;**************** CODE SEGMENT ****************************************

CODE    SEGMENT BYTE PUBLIC 'CODE'
        ASSUME  CS:CODE,ES:VARIABLEN,SS:STAPEL
ADC_OSZI        PROC    FAR

;BSAVE Simulation       (Umwandlung .EXE nach .BIN siehe Text)
        ORG     100H            ;SETZT OFFSET AUF 100
        DB      OFDH            ;BSAVEIDEN.
        DW      9040H           ;SEGMENTADRESSE
        DW      0100H           ;OFFSET
        DW      PROG_SIZE       ;LÄNGE DES PROGRAMMS
PROG_SIZE       EQU     OFFSET PROG_END -OFFSET PROG_BEG

;**************** PROGRAMM ********************************************

PROG_BEG:                       ;PROGRAMMBEGINN

COMMENT $ :  Den Beginn des Programms im Speicher (Offset) ermitteln, in
BX speichern und die Programmlänge+10H hinzuaddieren.  Somit steht jetzt
in  BX  der  Offset des 16.  Bytes nach dem Programm.  Division durch 16
ergibt die Zahl, die zum  CS  addiert  werden  muß,  um  das  Variablen-
Segment zu bestimmen (1.mögliche Segment nach dem Programm).
$
        CALL    GET_IP
        ADD     BX,PROG_SIZE+10H
        MOV     CL,4            ;Division durch 16
        SHR     BX,CL

        PUSH    BP
        MOV     BP,SP           ;mache Stack adressierbar

COMMENT  $  :Extrasegment  initialisieren,  Parameter  holen,  ins
Extrasegment schreiben. Zeit für Scalierung aus WAIT_TIME und Messung mit
Oszilloskop berechnen.  CS an BASIC übergeben.
$
        PUSH    ES              ;Extrasegment retten
INIT_SEGM       CS,BX,ES

GET_PARAM       P_WAIT_TIME,WAIT_TIME
        MOV     AX,TIME_MULT
        MUL     BX              ;AX = TIME_MULT*WAIT_TIME
        ADD     AX,TIME_DISP    ;AX = TIME_MULT*WAIT_TIME+TIME_DISP
        MOV     BX,10
        DIV     BX
        MOV     SCAL_TIME,AX    ;Zeitanzeige
GET_PARAM       P_CSEG,A_LO_DAT
PUT_PARAM       P_CSEG,CS

;Anfangswerte ins ES schreiben

;ADRESSEN
        MOV     AX,A_LO_DAT             ;Basisadresse
        INC     AX
        MOV     A_HI_DAT,AX
        MOV     A_STAT_EOC,AX           ;A_STAT_EOC=A_HI_DAT
        INC     AX
        MOV     A_STAT_START,AX

CHANGE_RANGE    A_9BIT,KORR_9,SCAL_9
        MOV     MIN_MAX_MODE_A,0
        MOV     MIN_MAX_MODE_B,1
        MOV     SIN_STP_MODE_A,0
        MOV     SIN_STP_MODE_B,1
        MOV     AKKUMUL_MODE_A,0
        MOV     AKKUMUL_MODE_B,1
INIT_PAGE

;erstmaliges  Initialisieren  der  Herkulesgraphik

GRAPH_X         GMODE
GRAPH_X         CLRSCR
STRING_X        50,LT_LINE,<A,D,C,,I,N,,R,A,N,D,O,M,,M,O,D,E>

COMMENT  $  :   erstmaliges  Initialisieren  des ADC ist im Strobe Modus
nicht nötig.  Für jeden RUN muß LO_DAT neu initialisiert werden
$
        MOV     DX,A_LO_DAT     ;Adresse für Daten
        MOV     AL,INIT_DAT
        OUT     DX,AL           ;Datum wird geschrieben

COMMENT $ :Erster Einsprung Punkt ins Programm ist AKKUMUL (FIRST_BEAT),
damit  das  1.   Bild  sofort nach Umschalten in RANDOM Mode in Echtzeit
sichtbar wird.
$
        JMP     FIRST_BEAT
NEXT_BEAT:

;************** AKKUMUL_MODE TEST

        CMP     AKKUMUL_MODE_A,0
        JE      NEW_PAGES
        JMP     AKKUMUL

NEW_PAGES:

;************** SINGLE_STEP_MODE TEST

        CMP     SIN_STP_MODE_A,1
        JE      SINGLE_STEP_MODE_ON     ;Disp P =Graphik P =ONE_PAGE
```

```
XCHG_PAGE       AL,ONE_PAGE,OTHER_PAGE
GRAPH_PAGE      AL                        ;ONE_PAGE  = GPAGE
DISP_PAGE       OTHER_PAGE                ;Disp Seite =Graphik Seite+1

SINGLE_STEP_MODE_ON:

GRAPH_X         CLRSCR
STRING_X        20,LT_LINE,<S,,,S,I,N,G,L,E,,S,T,E,P>
        ADD     DI,40                     ;4 Leerzeichen
STRING_X        ,,<M,,,M,I,N,M,A,X,Y>
        ADD     DI,40                     ;4 Leerzeichen
STRING_X        ,,<B,,,B,E,R,E,I,C,H>
        ADD     DI,50                     ;5 Leerzeichen
STRING_X        ,,<A,,,A,K,K,U,M,U,L,>
        ADD     DI,30                     ;3 Leerzeichen
STRING_X        ,,<H,,,H,A,L,T,E,N>

FIRST_BEAT:
DRAW_AXES       0,GROUND
SCAL_Y_AXE      SCAL,GROUND,<PKT,Z2,Z5,M+SHT,V,SLH,D,I,V>,
SCAL_X_AXE      72,GROUND, <M+SHT,S+SHT,SLH,D,I,V>,
PRINT_4_NUMBERS X_MAX-10,GROUND-4,SCAL_TIME,4,

AKKUMUL:
        MOV     MAX_POINT,Y_MAX
        MOV     MIN_POINT,0

COMMENT  $  :  Beginn der Messung:  START ist Synchronisationsimpuls des
ADC Adresse des Startsignals A_STAT_START muß für jeden BEAT neu  in  DX
geladen werden.  Der ADC schreibt als START- Signal, eine 0 in B0 an die
Adresse  A_STAT_START.   Das  Signal  wird  beim  Lesen  vom  Computer
invertiert  und  somit  wird  eine  1  in B0 an der Adresse A_STAT_START
(=A_COMMAND) als START- Signal aufgefasst.
$
        CLI                     ;schaltet alle Interruptanforderungen aus
        MOV     DX,A_STAT_START         ;Adresse für START-Signal
START_?:
        IN      AL,DX           ;B0 = START Bit (=1)
        SHR     AL,1            ;B0 in CF geshiftet
        JNC     START_?         ;Test CF=0, dann Sprung

;****************** SAMP ******************************************

        XOR     DI,DI           ;SAMP-Zähler für jeden BEAT neu setzen

NEXT_SAMP:
        MOV     DX,A_HI_DAT     ;A_STAT_EOC = A_HI_DAT

COMMENT $ :  ADC läuft automatisch und muß nicht getriggert werden.  Vom
Test  auf START-Impulses bis zur ersten Abfrage nach EOC vergehen xx TZ,
dh xxxx µs. Die Abfrage nach EOC dauert 26 (+8=34)  Taktzykl.   dh  5,5
(+1.7=7.2) µs.  Der ADC schreibt als EOC- Signal (10µs), eine 0 in B7 an
die Adresse  A_STAT_EOC.   Das  Signal  wird  beim  Lesen  vom  Computer
invertiert  und  somit  wird  eine  1  in  B7  an der Adresse A_STAT_EOC
(=A_HI_DAT) als EOC- Signal aufgefasst.
$
EOC_?:                          ;EOC End Of Conversion
        IN      AL,DX           ;B7=EOC Bit (=1)
        SHL     AL,1            ;B7 in CF geshiftet
        JNC     EOC_?           ;Test CF=0, dann Sprung

COMMENT $ :  Daten einlesen und plotten, während  der  ADC  weiterläuft.
Die 4 HI_BITS (schon bei EOC_? eingelesen) sind in AL=BBBBxxxx
$
        XOR     AH,AH           ;löscht AH Register
        MOV     CL,4
        SHL     AX,CL

        DEC     DX              ;A_LO_DAT einstellen

        IN      AL,DX           ;LO_BYTE einlesen in AL
        DIV_2   AX,RANGE        ;Auflösung einstellen
                                ;AX:=  ..HHHHLLLLL..  9 bis 12 Bits

COMMENT $ :Koordinatentransformation:
1. Translation: Yt= Y_DAT-K_TRANS_x    2. Spiegelung: Ys=2*GROUND -Yt
es folgt: Y_KORR= KORR_x -Y_DAT
$
        MOV     BP,KORR
        SUB     BP,AX           ;Y-Koordinate (max 349)

        CMP     BP,MAX_POINT    ;Test auf max. Spannung (HOLD Fkt.)
        JGE     OLD_MAX_POINT
        MOV     MAX_POINT,BP

OLD_MAX_POINT:
        CMP     BP,MIN_POINT    ;Test auf min. Spannung (HOLD Fkt.)
        JLE     OLD_MIN_POINT
        MOV     MIN_POINT,BP

OLD_MIN_POINT:
GRAPH_X         PLOT    ;Plottroutine: SAMP-Zähler DI=X-Koordinate (max 719)

        MOV     CX,WAIT_TIME
WAIT_LABEL:     NOP
        LOOP    WAIT_LABEL

;alle Samplings?

        INC     DI
        CMP     DI,X_MAX
        JNE     NEXT_SAMP

;****************** SAMP ENDE **************************************

;****************** MODE TESTS *************************************

        CMP     MIN_MAX_MODE_A,0
        JE      LOOK_FOR_KEY
DRAW_X_LINE     MAX_POINT
DRAW_X_LINE     MIN_POINT

LOOK_FOR_KEY:
BIOS_KEY        KEY_LOOK        ;wenn Zeichen in Tastatur, dann ist ZF=0
        JNZ     FIRST_TEST      ;Sprung, wenn ZF=0
        JMP     NEXT_BEAT       ;Sprung, wenn ZF=1  (kein Zeichen)

;****************** TASTATURTESTS **********************************

COMMENT $ :Die folgenden Tastaturtests werden durchgeführt:  TEST
AUF HOLD, AUF SINGLE_STEP, AUF AKKUMULIEREN, AUF Y_HOLD_MODE, AUF RANGE
$
FIRST_TEST:
BIOS_KEY        KEY_TEST

;**************          TEST AUF HOLD

        CMP     AL,H            ;HOLD Mode, wenn H in Tastatur (ZF=0)
        JNE     SINGLE_STEP_?
GRAPH_PAGE      OTHER_PAGE
STRING_X        X_MAX-200,FT_LINE,<B,I,L,D,,S,T,E,H,T>

HOLD_?:                         ;Warten auf das Ende des Haltens
BIOS_KEY        KEY_LOOK        ;wenn Zeichen in Tastatur (ZF=0)
        JZ      HOLD_?          ;Sprung, wenn ZF=1  (kein Zeichen)
BIOS_KEY        KEY_TEST        ;Tastaturpuffer löschen
        JMP     NEXT_BEAT       ;Sprung, wenn ZF=1
```

```
COMMENT  $  :Im  SINGLE STEP ebenso wie im AKKUMUL- Mode werden nur eine
der beiden Graphik Seiten benutzt nämlich OTHER_PAGE
$
;***************        TEST AUF SINGLE_STEP

SINGLE_STEP_?:

        CMP     AL,S                ;SINGLE STEP Mode umschalten,
        JNE     AKKUMUL_?           ;wenn S in Tastatur
XCHG_VARIAB     AL,SIN_STP_MODE_A,SIN_STP_MODE_B
GRAPH_PAGE      OTHER_PAGE
DISP_PAGE       OTHER_PAGE
        JMP     NEXT_BEAT

;***************        TEST AUF AKKUMULIEREN

AKKUMUL_?:
        CMP     AL,A                ;AKKUMUL Mode, wenn A in Tastatur
        JNE     MIN_MAX_POINT_?
XCHG_VARIAB     AL,AKKUMUL_MODE_A,AKKUMUL_MODE_B
GRAPH_PAGE      OTHER_PAGE
DISP_PAGE       OTHER_PAGE
STRING_X        X_MAX-200,FT_LINE,<A,K,K,U,M,U,L>
        JMP     NEXT_BEAT

;***************        TEST AUF Y_HOLD_MODE

MIN_MAX_POINT_?:
        CMP     AL,M            ;YHOLD Mode umschalten, wenn M in Tastatur
        JNE     RANGE_?
XCHG_VARIAB     AH,MIN_MAX_MODE_A,MIN_MAX_MODE_B
        JMP     NEXT_BEAT

;***************        TEST AUF RANGE

RANGE_?:
        CMP     AL,B                ;Bereich umschalten, wenn B in Tastatur
        JE      NEW_RANGE_?
        JMP     GO_BASIC

NEW_RANGE_?:
GRAPH_PAGE      OTHER_PAGE          ;schaltet auf angezeigte Seite
STRING_X        610,FT_LINE,<A,U,F,L,O,E,S,U,N,G>
        ADD     BP,12
STRING_X        620,,<Z1,,,Z9,B,I,T>
        ADD     BP,12
STRING_X        620,,<Z2,,Z1,Z0,B,I,T>
        ADD     BP,12
STRING_X        620,,<Z3,,Z1,Z1,B,I,T>
        ADD     BP,12
STRING_X        620,,<Z4,,Z1,Z2,B,I,T>
;
LOOK_FOR_RANGE:
BIOS_KEY        KEY_LOOK            ;wenn Zeichen in Tastatur, dann ist ZF=0
        JZ      LOOK_FOR_RANGE      ;Sprung, wenn ZF=1

BIOS_KEY        KEY_TEST
        CMP     AL,Z1               ;9 Bit Bereich einstellen
        JNE     KEY_2_?
CHANGE_RANGE    A_9BIT,KORR_9,SCAL_9
        JMP     NEXT_BEAT

KEY_2_?:
        CMP     AL,Z2               ;10 Bit Bereich einstellen
        JNE     KEY_3_?
CHANGE_RANGE    A_10BIT,KORR_10,SCAL_10
        JMP     NEXT_BEAT
```

```
KEY_3_?:
        CMP     AL,Z3               ;11 Bit Bereich einstellen
        JNE     KEY_4_?
CHANGE_RANGE    A_11BIT,KORR_11,SCAL_11
        JMP     NEXT_BEAT

KEY_4_?:
        CMP     AL,Z4               ;12 Bit Bereich einstellen
        JNE     LOOK_FOR_RANGE
CHANGE_RANGE    A_12BIT,KORR_12,SCAL_12
        JMP     NEXT_BEAT

;****************** TASTATURTESTS  ENDE ****************************

;Wiederherstellen der BASIC Register und Parameter-Return

GO_BASIC:

GRAPH_X         TMODE
        POP     ES                  ;Extrasegment
        POP     BP                  ;Base-Pointer
        STI                         ;schaltet alle Interrupts ein
        RET     P_SIZE
ADC_OSZI        ENDP

;****************** SUBROUTINEN ************************************

;Den Inhalt des IP Registers zum Zeitpukt des Aufrufes von GET_IP
;ermitteln

GET_IP          PROC    NEAR
        POP     BX                  ;IP in BX schreiben
        PUSH    BX                  ;IP wieder auf Stack
        SUB     BX,3                ;BX auf IP bei Aufruf von GET_IP setzen
        RET
GET_IP          ENDP

;*********** SUBROUTINEN ENDE **************************************

PROG_END:                           ;ENDE DES PROGRAMMS
CODE    ENDS

        END                         ;ENDE DES PROGRAMMBLOCKS
```

```
%OUT HERK_G1.ASM        Autor M. Weiss  Dez 86

COMMENT $:      HERKULES GRAPHIK                VEREINBARUNGEN, MACROS

-- vor dem RUN muß HGC FULL (Bildschirminitialisierung für hochauflösende
Graphik) und INT10.COM (Änderung des BIOS Vektors) aufgerufen worden sein.
-- um Graphikseiten mit PRT SCR 1 bzw PRT SCR 2 zu drucken muß GRAPHICS,
und HARDCOPY PRINTER.DEF aufgerufen worden sein.

  spezielle Verwendung von Registern
        AL:     Parameter: Bufpage,
        AH:     Herkules Graphik Fkt.
        DI:     Herkules Übergaberegister für X-Koordinaten
        BP:     Herkules Übergaberegister für Y-Koordinaten
                (ACHTUNG: INT   10 zerstört u.U. alle Registerinhalte)
MACRO AUFRUFE:

GRAPH_X         G_FKT (AH)
                ruft die angewählte Graphik Fkt. auf
                Register: AH,
```

Bild 3. Die Grafik-Macro-Bibliothek HERK_G1

```
NUM_ASCII         NUM(AL),BASE()
                  Wandelt Zahl NUM in ihren ASCII-Code um, wenn BASE leer
                  bleibt bzw addiert BASE (zB NULL+1) zu NUM hinzu.
                  Register: AL
START_POINT       SPALTE(),ZEILE()
                  Der Macro läd die Register DI und BP mit Startkoordinaten.
                  Wird für SPALTE oder ZEILE nichts geschrieben, so werden
                  die aktuellen Werte von DI bzw BP genommen.
                  Register: BP, DI
STRING_X          SPALTE(),ZEILE(),<C1,C2,.....,Cn>
                  schreibt Textzeichen auf Bildschirm
                  Ci sind in ASCII Code (bzw vereinbart, ALPHABET.ASM)
                  Wird für SPALTE oder ZEILE nichts geschrieben, so werden
                  die aktuellen Werte von DI bzw BP genommen. Analog
                  bleiben Cj leer.
                  Register: AH, AL, BP, DI
PRINT_4_NUMBERS   SPALTE(),ZEILE(),NUM(AX),STELLEN(B_XYZ),BASE()
                  schreibt eine STELLENstellige Zahl rechtsbündig auf den
                  Bildschirm.  Wird für SPALTE oder ZEILE nichts
                  geschrieben, so werden die aktuellen Werte von DI bzw BP
                  genommen.  Wird für STELLEN nichts geschrieben, so wird
                  Zweistelligkeit angenommen.
                  Max (STELLEN)=4; Max (NUM) =2550 (wird getestet)
                  Register: AH, AL, BP, DI, BL
                  Speicher: B_XYZ (Byte)
PRINT_3_NUMBERS   SPALTE(),ZEILE(),NUM(AX),BASE()
PRINT_2_NUMBERS   SPALTE(),ZEILE(),NUM(AX),BASE()
DRAW_X_LINE       Y_POS (BP)
                  zeichnet X_Linie mit Höhe bei Y_POS
                  Register: AH, BP, DI
DRAW_Y_LINE       X_POS (DI)
                  zeichnet Y_Linie bei X_POS
                  Register: AH, BP, DI
DRAW_AXES         X_POS (DI),Y_POS (BP)
                  zeichnet Koordinatensystem mit Ursprung bei X_POS,Y_POS
                  Register: AH, BP, DI
SCAL_Y_AXE        DISTANCE,ZERO,<E,I,N,H,E,I,T>,SCAL_MODE()(..)
                  scaliert Y- Achse mit Nullpkt. bei (0;ZERO) und schreibt
                  Einheit an Position 30,10. Wenn SCAL_MODE=1, dann werden
                  zweistellige Zahlen geschrieben, wenn SCAL_MODE blank ist,
                  dann wird dieser Teil nicht assembliert, dh Zahlbeschrift=
                  ung ist unmöglich.
                  Register: AH, AL, BP, DI
                  Speicher: B_SCALE (Byte)
SCAL_X_AXE        DISTANCE,ZERO,<E,I,N,H,E,I,T>,SCAL_MODE()(..)
                  scaliert X- Achse mit Nullpkt. bei (0;ZERO) und schreibt
                  Einheit  an Position (530;ZERO+3) Wenn SCAL_MODE=1, dann
                  werden zweistellige Zahlen geschrieben,  wenn  SCAL_MODE
                  blank  ist,  dann wird dieser Teil nicht assembliert, dh
                  Zahlbeschriftung ist unmöglich.
                  Register: AH, AL, BP, DI
                  Speicher: B_SCALE (Byte)
INIT_PAGE
                  Initialisiert die beiden Variablen für die Graphikseitenzahl
                  Speicher: ONE_PAGE, OTHER_PAGE
XCHG_PAGE         REG,ONE_PAGE(REG),OTHER_PAGE
                  Die beiden Seiten ONE_PAGE und OTHER_PAGE werden gegenein=
                  ander vertauscht (0/1). Nach Ende des Macros steht in REG
                  dieselbe Zahl wie in ONE_PAGE.
                  Register: REG
                  Speicher: ONE_PAGE, OTHER_PAGE
DISP_PAGE         PAGE_? (AL)
                  Schaltet DISP-Page auf PAGE_?
                  Register: AL, AH
                  Speicher: (PAGE_?)
GRAPH_PAGE        PAGE_? (AL)
                  Schaltet GPAGE auf PAGE_?
                  Register: AL, AH
                  Speicher: (PAGE_?)
XCHG_GDPAGE
                  Die beiden Seiten ONE_PAGE und OTHER_PAGE werden gegenein=
                  ander vertauscht (0/1). Der Macro schaltet dann GPAGE auf
                  ONE_PAGE, und DISP-Page auf OTHER_PAGE.
                  Nach Ende des Macros sind die Seitenzahlen vertauscht.
                  Register: AL, AH
                  Speicher: ONE_PAGE, OTHER_PAGE
$
;************* VEREINBARUNGEN ******************************************

                  ;HERKULES-GRAPHIK

        ;ATTRIBUTFKT.
GMODE             EQU      40H      ;Graphik Modus
TMODE             EQU      41H      ;Text     "
GPAGE             EQU      43H      ;aktive Seite
DISP              EQU      45H      ;gezeigte  "
LEVEL             EQU      44H      ;Intensität
        ;ZEICHENFKT.  auf aktiver Seite
CLRSCR            EQU      42H      ;löscht Bildschirm
PLOT              EQU      46H      ;zeichnet Pkt
MOVE_CURS         EQU      48H      ;bewegt Curser
MOVE              EQU      48H      ;   "
DLINE             EQU      49H      ;zeichnet Linie von Curs. Pos. zu Pkt
CIRC              EQU      4DH      ;zeichnet Kreis
ARC               EQU      4CH      ;
BLKFIL            EQU      4AH      ;zeichnet Rechteck
FILL              EQU      4EH      ;füllt Fläche
TEXT              EQU      4BH      ;schreibt Zeichen
GETPT_INTENS      EQU      47H      ;liest Intensität
GETPT             EQU      47H      ;   "
        ;SPEZIELLE PARAMETER
WHITE             EQU      1        ;Intensitäten
BLACK             EQU      0
OTHER             EQU      2
PAGE_0            EQU      0        ;Seiten
PAGE_1            EQU      1
        ;BILDSCHIRM
X_MAX             EQU      719
Y_MAX             EQU      347
M_LINE            EQU      (Y_MAX+1)/2
MT_LINE           EQU      (Y_MAX+1)/2      ;mittlere Textlinie
LT_LINE           EQU      343              ;letzte Textlinie
FT_LINE           EQU      10               ;erste      "
IF1
;************* MACRO-VEREINBARUNGEN ************************************

GRAPH_X           MACRO             G_FKT
        IFDIF     <G_FKT>,<AH>
        MOV       AH,G_FKT          ;;     Graphik-Fkt in Befehlsregister
        ENDIF
        INT       10H               ;;     INT10.COM
ENDM
;-----------------------------------------------------------------------
NUM_ASCII         MACRO             NUM,BASE
        IFDIF     <NUM>,<AL>
        MOV       AL,BYTE PTR NUM
        ENDIF
        IFB       <BASE>
        ADD       AL,30H            ;;NUM+NULL == ASCII
        ELSE
        ADD       AL,BASE
        ENDIF
ENDM
```

```
;------------------------------------------------------------
START_POINT     MACRO           SPALTE,ZEILE
        IFNB    <ZEILE>
        MOV     BP,ZEILE        ;;      Koordinaten
        ENDIF
        IFNB    <SPALTE>
        MOV     DI,SPALTE       ;;
        ENDIF
ENDM
;------------------------------------------------------------
;************** Zeichen schreiben

STRING_X        MACRO           SPALTE,ZEILE,CHARS
        START_POINT     SPALTE,ZEILE    ;;Startkoordinaten

IRP     CHAR,<CHARS>            ;;      Zeichenliste
        IFNB    <CHAR>          ;;      bei "blank" gleich zu ADD
        MOV     AL,CHAR         ;;
        GRAPH_X         TEXT    ;;      Graphik-Fkt (TEXT)
        ENDIF
        ADD     DI,10           ;;      neue Spalte
ENDM
ENDM
;************** Zahlen schreiben

PRINT_4_NUMBERS MACRO           SPALTE,ZEILE,NUM,STELLEN,BASE
                LOCAL   NEXT_FIGURE, LAST_FIGURE, OVERFLOW_BREAK, NO_NUM

        START_POINT     SPALTE,ZEILE    ;;Startkoordinaten

        IFDIF   <NUM>,<AX>
        MOV     AX,WORD PTR NUM ;;Zahl, die gedruckt werden soll
        ENDIF

        IFNB    <STELLEN>
        CMP     AX,2550         ;;Quotient darf maximal 255 betragen,
        JG      OVERFLOW_BREAK  ;;da sonst AL -Register überläuft und ein
                                ;;INT 0 (Devide by Zero Interrupt) erfolgt
        IFDIF   <STELLEN>,<B_XYZ>       ;;Anzahl der Stellen
        MOV     B_XYZ,BYTE PTR STELLEN
        ENDIF
        ENDIF

NEXT_FIGURE:
        MOV     BL,10
        DIV     BL              ;;AX/BL --> Quot. in AL, Rest in AH
        PUSH    AX              ;;Ergebnis retten
        MOV     AL,AH           ;;Rest in AL schreiben
NUM_ASCII       AL,BASE         ;;Rest. in  ASCII umrechnen
GRAPH_X         TEXT            ;;und an Bildschirm ausgeben
        SUB     DI,10           ;;nächste (höherwertige) Stelle einstellen
        POP     AX              ;;Quot. holen (steht in AL)
IFNB    <STELLEN>
XOR     AH,AH           ;;AX=0000[AL]
DEC     B_XYZ           ;;Stellenzähler
CMP     B_XYZ,1         ;;nur noch eine Stelle ?
JG      NEXT_FIGURE     ;;nein ? dann nochmal !
ENDIF

CMP     AL,9            ;;Test auf Stellen Overflow (falls STELLEN<4)
JG      OVERFLOW_BREAK
CMP     AL,0            ;;Test auf vorlaufende Null und Negativität
JL      OVERFLOW_BREAK
JG      LAST_FIGURE
MOV     AL,SPC
JMP     SHORT NO_NUM
OVERFLOW_BREAK:
        MOV     AL,10H          ;;geschrieben wird dann @
LAST_FIGURE:
NUM_ASCII       AL,BASE         ;;die höchstwertige Stelle schreiben
NO_NUM:
GRAPH_X         TEXT
ENDM
;------------------------------------------------------------
PRINT_3_NUMBERS MACRO           SPALTE,ZEILE,NUM,BASE
        PRINT_4_NUMBERS         SPALTE,ZEILE,NUM,3,BASE
ENDM
;------------------------------------------------------------
PRINT_2_NUMBERS MACRO           SPALTE,ZEILE,NUM,BASE
        PRINT_4_NUMBERS         SPALTE,ZEILE,NUM,,BASE
ENDM

;************** Graphik Seiten

INIT_PAGE       MACRO
        MOV     ONE_PAGE,PAGE_0
        MOV     OTHER_PAGE,PAGE_1
ENDM
;------------------------------------------------------------
XCHG_PAGE       MACRO           REG,ONE_PAGE,OTHER_PAGE
        IFDIF   <ONE_PAGE>,<REG>
        MOV     REG,ONE_PAGE
        ENDIF
        XCHG    REG,OTHER_PAGE
        MOV     ONE_PAGE,REG
ENDM
;------------------------------------------------------------
DISP_PAGE       MACRO           PAGE_?
        IFDIF   <PAGE_?>,<AL>
        MOV     AL,PAGE_?
        ENDIF
GRAPH_X         DISP
ENDM
;------------------------------------------------------------
GRAPH_PAGE      MACRO           PAGE_?
        IFDIF   <PAGE_?>,<AL>
        MOV     AL,PAGE_?
        ENDIF
GRAPH_X         GPAGE
ENDM
;------------------------------------------------------------
XCHG_GDPAGE     MACRO
XCHG_PAGE       AL,ONE_PAGE,OTHER_PAGE
GRAPH_PAGE      AL              ;;ONE_PAGE
DISP_PAGE       OTHER_PAGE
ENDM

;************** Koordinaten

DRAW_X_LINE     MACRO           Y_POS
        XOR     DI,DI           ;;      Zeichnen der X- Linie
        IFDIF   <Y_POS>,<BP>
        MOV     BP,Y_POS
        ENDIF
        GRAPH_X         MOVE_CURS
        MOV     DI,X_MAX
        GRAPH_X         DLINE
ENDM
;------------------------------------------------------------
DRAW_Y_LINE     MACRO           X_POS
        XOR     BP,BP           ;;      Zeichnen der Y- Linie
        IFDIF   <X_POS>,<DI>
```

```
        MOV     DI,X_POS
        ENDIF
        GRAPH_X         MOVE_CURS
        MOV     BP,Y_MAX
        GRAPH_X         DLINE
ENDM
;------------------------------------------------------------------------
DRAW_AXES       MACRO           X_POS,Y_POS
        DRAW_Y_LINE     X_POS
        DRAW_X_LINE     Y_POS
ENDM
;------------------------------------------------------------------------
SCAL_X_AXE      MACRO           DISTANCE,ZERO,EINHEIT,SCAL_MODE
                LOCAL   POS_SCAL,NO_SCAL_P
        IFNB    <SCAL_MODE>
        MOV     B_SCALE,0
        ENDIF

        XOR     DI,DI
POS_SCAL:
        ADD     DI,DISTANCE
        MOV     BP,ZERO
        SUB     BP,3
        GRAPH_X         MOVE_CURS
        ADD     BP,6
        GRAPH_X         DLINE

        IFNB    <SCAL_MODE>
        INC     B_SCALE
        CMP     SCAL_MODE,1
        JNE     NO_SCAL_P
        NUMBER_X        ,,B_SCALE,
        SUB     DI,10
NO_SCAL_P:
        ENDIF

        CMP     DI,X_MAX
        JL      POS_SCAL

        IFNB    <EINHEIT>
        ADD     BP,10
        STRING_X        660,,<EINHEIT>
        ENDIF
ENDM
;------------------------------------------------------------------------

SCAL_Y_AXE      MACRO           DISTANCE,ZERO,EINHEIT,SCAL_MODE
                LOCAL   NEG_SCAL,POS_SCAL,NO_SCAL_P,NO_SCAL_N
        IFNB    <SCAL_MODE>
        MOV     B_SCALE,0
        ENDIF

        MOV     BP,ZERO
NEG_SCAL:
        XOR     DI,DI
        ADD     BP,DISTANCE
        GRAPH_X         MOVE_CURS
        MOV     DI,10
        GRAPH_X         DLINE
        IFNB    <SCAL_MODE>
        INC     B_SCALE
        CMP     SCAL_MODE,1
        JNE     NO_SCAL_N
        NUMBER_X        ,,B_SCALE,
NO_SCAL_N:
        ENDIF
        CMP     BP,Y_MAX
        JL      NEG_SCAL
        IFNB    <SCAL_MODE>
        MOV     B_SCALE,0
        ENDIF

        MOV     BP,ZERO
POS_SCAL:
        XOR     DI,DI
        SUB     BP,DISTANCE
        GRAPH_X         MOVE_CURS
        MOV     DI,10
        GRAPH_X         DLINE
        IFNB    <SCAL_MODE>
        INC     B_SCALE
        CMP     SCAL_MODE,1
        JNE     NO_SCAL_P
        NUMBER_X        ,,B_SCALE,
NO_SCAL_P:
        ENDIF

        CMP     BP,0
        JG      POS_SCAL
        IFNB    <EINHEIT>
        STRING_X        30,10,<EINHEIT>
        ENDIF
ENDM

;************* MACRO-VEREINBARUNGEN  ENDE ******************************

ENDIF
```

```
%OUT ADC_MAC1.ASM       Autor M. Weiss  Dez 86

COMMENT $ :
-- für einige Macros sollte ALPHABET.ASM included werden

Macro Aufrufe:  Kurzübersicht

GET_PARAM       P_NAME,V_NAME
PUT_PARAM       P_NAME,V_NAME (BX)
INIT_SEGM       OLD_SEG (AX),DELAY,NEW_SEG (AX)

Prozessorbefehle auf 2 Speichervariablen erweitert
XCHG_VARIAB     REG,VARIAB_1(REG),VARIAB_2
MOV_VARIAB      REG,VARIAB_1,VARIAB_2(REG)
CMP_VARIAB      REG,VARIAB_1(REG),VARIAB_2

Prozessorbefehle variiert
DIV_2           REG,DIVISOR
MUL_2           REG,MULTIPLICATOR
LOOP_WAIT       WAIT_TIME (CX)

BIOS_KEY        K_FKT (AH)
GET_NUMBER      NUM(AX)

DOS_FKT         D_FKT (AH)
DOS_STRING      <C1,C2,.....Cn>
$

;************* VEREINBARUNGEN *****************************************

                ;BIOS/DOS
KEY_TEST                EQU     0
KEY_LOOK                EQU     1       ;Zeichen im Puffer ?
```

Bild 4. Die Macro-Bibliothek ADC_MAC1

```
PRINT_CHAR          EQU     2
DOS                 EQU     21H     ;DOS Fkt. Dispatcher

IF1         ;Macrodefinitionen werden nur beim 1. Lauf included

;************* MACRO-VEREINBARUNGEN ********************************

GET_PARAM       MACRO           P_NAME,V_NAME
        MOV     BX,[BP].&P_NAME ;;Hole Adresse von NAME
        MOV     BX,[BX]
        MOV     V_NAME,BX       ;;NAME in Speicher schreiben
ENDM
;-------------------------------------------------------------------
PUT_PARAM       MACRO           P_NAME,V_NAME
        IFDIF   <V_NAME>,<BX>
        MOV     BX,V_NAME       ;;NAME der zu Überg. Variablen
        ENDIF
        MOV     DI,[BP].&P_NAME ;;Adresse des Rückgabeparam.
        MOV     [DI],BX         ;;CS in diese Adresse schreiben
ENDM
;-------------------------------------------------------------------
INIT_SEGM       MACRO           OLD_SEG,DELAY,NEW_SEG
        IFDIF   <OLD_SEG>,<AX>
        MOV     AX,OLD_SEG
        ENDIF
        ADD     AX,DELAY
        IFDIF   <NEW_SEG>,<AX>
        MOV     NEW_SEG,AX      ;;Segment initialisieren
        ENDIF
ENDM
;Prozessorbefehle auf 2 Speichervariablen erweitert

XCHG_VARIAB     MACRO           REG,VARIAB_1,VARIAB_2
        IFDIF   <VARIAB_1>,<REG>
        MOV     REG,VARIAB_1
        ENDIF
        XCHG    REG,VARIAB_2
        MOV     VARIAB_1,REG
ENDM
;-------------------------------------------------------------------
MOV_VARIAB      MACRO           REG,VARIAB_1,VARIAB_2
        IFDIF   <VARIAB_2>,<REG>
        MOV     REG,VARIAB_2
        ENDIF
        MOV     VARIAB_1,REG
ENDM
;-------------------------------------------------------------------
CMP_VARIAB      MACRO   REG,VARIAB_1,VARIAB_2
        IFDIF   <VARIAB_1>,<REG>
        MOV     REG,VARIAB_1
        ENDIF
        CMP     REG,VARIAB_2
ENDM
;-------------------------------------------------------------------
;Prozessorbefehle variiert

DIV_2           MACRO           REG,DIVISOR
                LOCAL   NO_DIVISION
        MOV     CL,BYTE PTR DIVISOR
        JZ      NO_DIVISION
        SHR     REG,CL
NO_DIVISION:
ENDM
;-------------------------------------------------------------------
MUL_2           MACRO           REG,MULTIPLICATOR
                LOCAL   NO_MULTIPLICATION
        MOV     CL,BYTE PTR MULTIPLICATOR
        JZ      NO_MULTIPLICATION
        SHL     REG,CL
NO_MULTIPLICATION:
ENDM
;-------------------------------------------------------------------
LOOP_WAIT       MACRO           WAIT_TIME
                LOCAL   WAIT_LABEL, NO_WAIT_LABEL
        IFDIF   <WAIT_TIME>,<CX>
        MOV     CX,WAIT_TIME
        ENDIF
WAIT_LABEL:
        NOP
        LOOP    WAIT_LABEL
ENDM
;-------------------------------------------------------------------

;BIOS Funktionen (INT 16)

BIOS_KEY        MACRO           K_FKT
        IFDIF   <K_FKT>,<AH>
        MOV     AH,K_FKT        ;;Befehlsregister laden
        ENDIF
        INT     16H
ENDM
;-------------------------------------------------------------------
GET_NUMBER      MACRO   NUM
                LOCAL   LOOK_FOR_NR
LOOK_FOR_NR:
BIOS_KEY        KEY_LOOK        ;wenn Zeichen in Tastatur, dann ist ZF=0
        JZ      LOOK_FOR_NR     ;Sprung, wenn ZF=1

BIOS_KEY        KEY_TEST
        CMP     AL,NULL
        JL      LOOK_FOR_NR
        CMP     AL,Z9
        JG      LOOK_FOR_NR

        SUB     AL,NULL
        XOR     AH,AH
        IFDIF   <NUM>,<AX>
        MOV     NUM,AX
        ENDIF
ENDM
;-------------------------------------------------------------------
;DOS Function Dispatcher (INT 21)

DOS_FKT         MACRO           D_FKT
        IFDIF   <D_FKT>,<AH>
        MOV     AH,D_FKT        ;;Befehlsregister laden
        ENDIF
        INT     21H
ENDM
;-------------------------------------------------------------------
DOS_STRING      MACRO           CHARS
IRP     CHAR,<CHARS>
        IFB     <CHAR>
        MOV     DL,SPC
        ELSE
        MOV     DL,CHAR
        ENDIF
        DOS_FKT         PRINT_CHAR
ENDM
ENDM

;************* MACRO-VEREINBARUNGEN ENDE ****************************

ENDIF
```

Dr. rer. nat. Michael Bach

Batch-Dateien mit mehr Komfort

Wenn man von CP/M+ zu MS-DOS wechselt, stellt man einige Nachteile von MS-DOS fest. Ein gravierender Nachteil von MS-DOS besteht darin, daß man von Batch-Dateien aus keine direkten Programmeingaben vornehmen kann. Ein kleines Turbo-Pascal-Programm löst das Problem.

In CP/M+ ist das kein Problem: Wenn eine Zeile in einer Submit-Datei mit „<“ beginnt, werden alle nachfolgenden Zeichen, als ob sie von der Tastatur kämen, dem gerade aktiven Programm übergeben. Das ist sehr praktisch bei Programmen, die umfangreiche Eingaben benötigen, aber auch automatisch funktionieren sollen. Unter MS-DOS kann eine Batch-Datei zwar Programme aufrufen, ihnen dann aber nur Kommandozeilen-Parameter übergeben. Auch die Eingangsumleitung (Input Redirection) hilft nicht viel weiter. Kommandozeilen-Parameter, wie sie z. B. der Microsoft-Makro-Assembler (MASM) kennt, sind unpraktisch und nicht bei allen Programmen implementiert. Und Redirektion ist zwar im Prinzip mächtig, hat jedoch folgende Nachteile:

- Man muß dafür eine zweite Datei erstellen. Damit wird die Sache etwas unübersichtlich.
- Es passiert Schreckliches, wenn die Datei zu Ende ist, von der anstelle der Tastatureingaben gelesen wird. Das System hängt sich auf!

Als typisches Beispiel sei Turbo-Pascal genannt:

Wie praktisch wäre es, wenn man „TP <NAME>“ schreiben könnte, und schon startete durch den Batch TP der Turbo-Pascal-Compiler, die Frage nach der Fehler-Datei würde mit „y“ beantwortet, mit „e“ der Editor gewählt und die Datei <NAME> eingelesen!

Direktes Schreiben in den BIOS-Tastatur-Puffer

Das Turbo-Pascal-Programm TBATCH verwendet eine sehr einfache Methode, um kurze Programmeingaben umzulenken. Der Trick liegt darin, die Programmeingaben direkt in den Tastatur-Puffer des BIOS zu schreiben. Im Tastatur-Puffer haben bis zu 15 Zeichen Platz, also genug, um Turbo-Pascal wie gewünscht zu starten. Wenn der Turbo-Pascal-Compiler nach Eingaben fragt, übergibt ihm das BIOS die Zeichen vom Tastatur-Puffer, ohne zu ahnen, daß die nicht auf normalem Weg hineingekommen sind. Entscheidend für die Verwaltung des Tastatur-Puffers sind die BIOS-Variablen BufferHead und BufferTail, die nach Art eines Ringpuffers die Warteschlange verwalten. Wenn der Wert der Variablen BufferHead dem Wert der Variablen BufferTail gleicht, ist der Puffer voll. Da kein echter Ringpuffer, sondern ein linearer Puffer vorliegt, müssen die Variablen BufferHead und BufferTail wieder auf ihre Anfangswerte gesetzt werden, wenn der Pufer voll ist. Die Adressen der Pufferzeiger und die Adresse

```
rem Start von Turbo-Pascal 19.10.87 MB

tbatch ye%1\n                  ; "ye<NAME><CR><LF>" in den Puffer
turbo                          ; Turbo liest den Puffer wieder aus
```

Bild 1. Tippen Sie "tp NAME" und mit dieser Batch-Datei TP wird Turbo-Pascal gestartet und die Datei NAME.PAS in den Turbo-Editor geladen

```
program TBatch;

{ Simulation kurzer Tastatureingaben für Batch-Anwendungen }

type str80=string[80];

var
  p: integer; s: str80;
  BufferHead: integer absolute $40:$1a;
  BufferTail: integer absolute $40:$1c;


function NextChar: char;
begin
  p:=succ(p);
  if p<=length(s) then NextChar:=s[p] else NextChar:=' ';
end;


procedure PutChar(c: char);
begin
  inline($fa); { Disable interrupts }
  memw[$40:BufferTail]:=ord(c); BufferTail:=BufferTail+2;
  if BufferTail>=$3E then BufferTail:=$1E;
  inline($fb{sti}); { Enable Interrupts }
end;
```

Bild 2. Kurze Tastatureingaben simuliert dieses Turbo-Pascal-Programm

```
procedure Eintragen(s: str80);
var c: char;
begin
  p:=0;
  while p<length(s) do begin
    c:=NextChar;
    if c='\' then begin
      case UpCase(NextChar) of
        'N': begin PutChar(^M); PutChar(^J); end;
        'B': PutChar(^H);
        'T': PutChar(^I);
        '\': PutChar('\');
        ' ': PutChar(' ');
      end;
    end else PutChar(c);
  end;
end; { Eintragen }

begin  { Haupt }
  if ParamCount<1 then begin
    writeln(^G,'Kein Paramter angegeben!'); exit;
  end;
  s:='';
  for p:=1 to ParamCount do s:=s+' '+Paramstr(p);
  delete(s,1,1);
  if length(s)<15 then Eintragen(s) else
    writeln(^G,'Zeichenkette zu lang!');
end.
```

des Puffers kann man dem BIOS-Listing des jeweiligen PCs entnehmen.

Interrupts sind nicht zugelassen

Es ist wichtig, daß das Programm die Interrupts abschaltet, bevor es die Pufferzeiger setzt; sonst gibt es einen Konflikt, wenn im gleichen Augenblick ein Tastatur-Interrupt seinerseits die Zeiger verändert. Das Programm ist sehr primitiv und überprüft noch nicht einmal einen Pufferüberlauf. Der tritt praktisch nicht auf, da der Puffer automatisch leer ist, wenn man den Batch-Lauf (Beispiel TP) manuell startet. Nach Abarbeitung der übergebenen Zeichenkette erfolgt die Eingabe wieder normal über die Tastatur.

Die Batch-Datei TP zum Starten von Turbo-Pascal ist in *Bild 1* dargestellt.

Das zu ladende Programm wird als Parameter hinter TP angegeben und für „%1" eingesetzt. Das mysteriöse „\n" steht für die Eingabe der Enter- bzw. Carriage-Return-Taste und ist der Syntax von C abgeschaut: Das „\" ist ein Fluchtsymbol (Escape), d. h. das nachfolgende Zeichen wird uminterpretiert. Das Programm TBATCH wandelt „\n" zu <CR><LF>, „\b" zu <BackSpace>, „\\" zu „\" und „\ " zu " " um. Ein Backslash (\) für Pfade einzugeben, ist sehr einfach. Man muß einfach den Backslash zweimal eingeben.

PUTCHAR und GETCHAR wandeln Escape-Sequenzen um

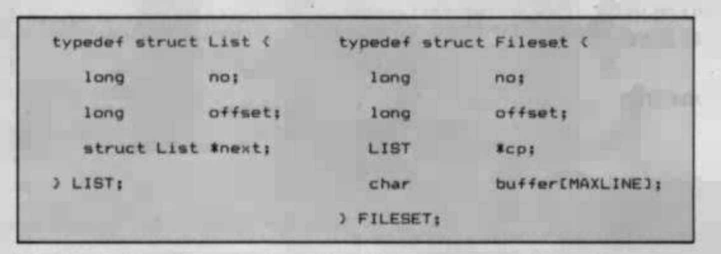
Die im Programm vielleicht etwas aufwendig erscheinenden Umwege über PUTCHAR und GETCHAR vereinfachen die Umwandlung von Escape-Sequenzen verschiedener Länge. Das Programm (*Bild 2*) kommt ohne im Speicher wohnende Teile aus und es „biegt" auch keinen Interrupt-Vektor um. So praktisch und simpel das Programmchen auch ist, seine Nachteile seien nicht verschwiegen:

- Das Programm geht von festen BIOS-Adressen für den Puffer aus. Das ist zwar nicht so schlimm wie das direktes Aufrufen von BIOS-Routinen, aber auch nicht portabel.
- Die freie Platz im Puffer wird nicht geprüft.

Das wäre zwar leicht einzubauen, aber was soll das Programm tun, wenn der Platz im Puffer zu klein ist? Wer wirklich viele direkte Programmeingaben umlenken will, dem sei EBL (Extended Batch Language) empfohlen. Wer mit dem Norton-Commander arbeitet, sollte „TP !.PAS" in die Extensionen-Datei eintragen. Dann kann man mit der Maus den Programm-Namen doppelklicken, und schon ist man im Turbo-Editor – fast wie beim Macintosh. □

C-Compiler unter CP/M-68k

Alle, die mit dem C-Compiler von Digital-Research intensiver arbeiten möchten, seien an dieser Stelle auf eine „Falle" hingewiesen: Diese resultiert anscheinend aus der sehr strengen Anlehnung an den Kernighan/Ritchie-Standard; viele jüngere C-Compiler erweisen sich hier als toleranter. Zum Problem: Wenn in zwei Strukturen gleichnamige Strukturelemente an verschiedenen Stellen auftreten und im Programm Zuweisungen zwischen diesen Strukturelementen erfolgen, dann ist nicht eindeutig, auf welches Strukturelement sich der Compiler bezieht. In einer C-Quelle stehen beispielsweise die in *Bild 1* gezeigten Typen-Definitionen. Werden diese Typen nun vom Programm so benutzt, wie es die Funktion in *Bild 2* macht, dann besteht für den Compiler das Problem darin, zu erkennen, daß das Strukturelement „offset" in der zweiten Zuweisung auf der linken Seite zum Typ „Fileset" und auf der rechten Seite zum Typ „List" gehört. Leider gibt der Compiler hier nicht etwa eine Warnung aus, sondern behandelt die Zuweisung so, als gehörte das Strukturelement „offset" auf beiden Seiten zum gleichen Typ. Tritt dagegen „offset" nur einmal in der Zuweisung auf, so erzeugt der Compiler korrekten Code. Dies ist z. B. durch Zwischenspeicherung möglich *(Bild 3)*. Will man solch unschöne Klimmzüge vermeiden und trotzdem dem Kernighan/Ritchie-Standard entsprechen, so muß man dafür sorgen, daß gleichnamige Strukturelemente in verschiedenen Strukturen an gleicher Stelle stehen, typgleich sind und typgleiche Vorgänger in der Struktur haben. Diese Anforderungen sind in dem Buch „Programmieren in C" im Anhang A 8.5 auf Seite 218 beschrieben. Eine Lösung für das vorgestellte Beispiel zeigt *Bild 4*. Der Grund für dieses nicht sofort verständliche Verhalten liegt wohl darin, daß die Namen von Strukturelementen in C nicht an die Struktur gebunden sind. *Harald Halm/Olaf Zurth*



```
typedef struct List {          typedef struct Fileset {
    long    no;                    long    no;
    long    offset;                long    offset;
    struct List *next;             LIST    *cp;
} LIST;                            char    buffer[MAXLINE];
                               } FILESET;
```

Bild 1. Die Strukturvariable „offset" ist zwar vom gleichen Typ, steht aber an unterschiedlicher Stelle in der Struktur

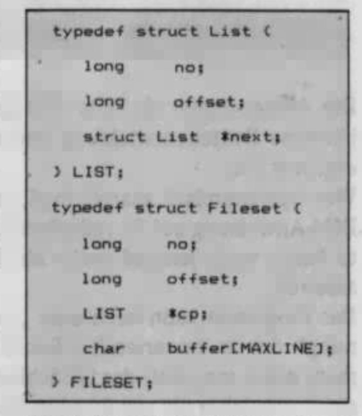
```
int Match()
{
LIST      *cp;
FILESET   *fsa, *fsb;               /* weitere Deklarationen */
                    /* ...irgendwas... */
cp = fsb->cp;
fsa->offset = cp->next->offset; /* hier passiert's */
                    /* ...irgendwas... */
}
```

Bild 2. Über Zeiger auf die Strukturen wird auf die beiden gleichnamigen Variablen zugegriffen

```
long   dummy;      /* Hilfsvariable */

dummy = cp->next->offset;
fsa->offset = dummy;
```

Bild 3. So erzeugt der Compiler richtigen Code – ein mühsamer Weg, der auch die Laufzeit des Programms verlängert



```
typedef struct List {
    long   no;
    long   offset;
    struct List  *next;
} LIST;
typedef struct Fileset {
    long   no;
    long   offset;
    LIST   *cp;
    char   buffer[MAXLINE];
} FILESET;
```

Bild 4. So sieht die richtige Lösung aus: Nicht nur der Typ der Strukturvariablen selbst, sondern auch Typ und Anzahl der vorangehenden Variablen sind gleich

Rentenberechnung für jedermann

Besonders die jüngeren mc-Leser denken wohl kaum an ihre Rente. Dennoch ist auch für diese das Programm interessant, da es eine Hochrechnung der zu erwartenden Rente vornimmt. Für die Berechnung der Rente und den Ausdruck der Rentenentwicklung sind wenige Angaben erforderlich.

Das Programm zur Rentenberechnung (*Bild*) ist in GW-Basic geschrieben. Alle erforderlichen Tabellenwerte sind bereits in ihm enthalten. Der wichtigste Teil ist der DATA-Block in den Zeilen 110 bis 160. Hier sind alle Tabellenwerte für die Berechnung der Rente ab 1942 abgelegt. Diese Zeilen können in den nächsten Jahren erweitert werden, damit das Programm aktuell bleibt.

Die Rentenberechnung im Programm beginnt mit dem Vergleich des Tabellenwerts FE() – dies ist der Durchschnittverdienst aller Beschäftigten für ein bestimmtes Jahr – mit dem Bruttoverdienst im entsprechenden Jahr (Zeile 300, INPUT BE(I)). Hieraus errechnet sich der individuelle Faktor B(I) in Prozent.

In der Zeile 70 werden drei verschiedene Felder angelegt. Das Feld FE() beinhaltet die oben erwähnten Tabellenwerte, BE() ist der Bruttoverdienst im entsprechenden Jahr laut Versicherungsnachweis, B() ist der Bemessungsfaktor in Prozent für das jeweilige Jahr.

Die Formel zur Berechnung der Monatsrente lautet:

> Rente =
> (Versicherungsjahre × persönlicher Bemessungsfaktor × Rentenbemessungsgrundbetrag × 1,5%)/12

Im Basic-Programm steht daher in Zeile 410 die folgende Anweisung:

> RE = BJ × BP × BA × 1.5 / 1200

Die Anweisungen ab Zeile 460 geben die jährliche Rentenentwicklung und das Endergebnis aus.

Wer Speicherplatz sparen muß, kann die DIM-Anweisung auf 50 reduzieren, da heute kaum noch jemand mehr als 50 Jahre arbeitet.

Das Programm kann auch eine „Hochrechnung" der zu erwartenden Rente vornehmen, wenn man statt der tatsächlichen Versicherungsjahre die gewünschte Anzahl von Versicherungsjahren bei der Frage „Für wieviele Jahre berechnen?" eingibt. Es berechnet auf der Basis von 1986 die zu erwartende Rente.

Für die Berechnung der Rente und den Ausdruck der Rentenentwicklung sind folgende Angaben erforderlich:

- Jährlicher Bruttoverdienst seit dem Beginn der Berufstätigkeit
- Eventuelle Ausbildungs- oder Ersatzzeiten vor 1965
- Eventuelle Ausbildungs- oder Ersatzzeiten seit 1965

Zu beachten ist, daß jede Zeitangabe als Dezimalzahl eingegeben werden muß (statt 12½ Jahre – 12.50 Jahre!). Ansonsten ist das Programm weitgehend selbsterklärend. Jeder, der ab 1942 berufstätig wurde, kann hiermit seine Rente selbst errechnen, mit einer Abweichung von 1...5 DM/Monat.

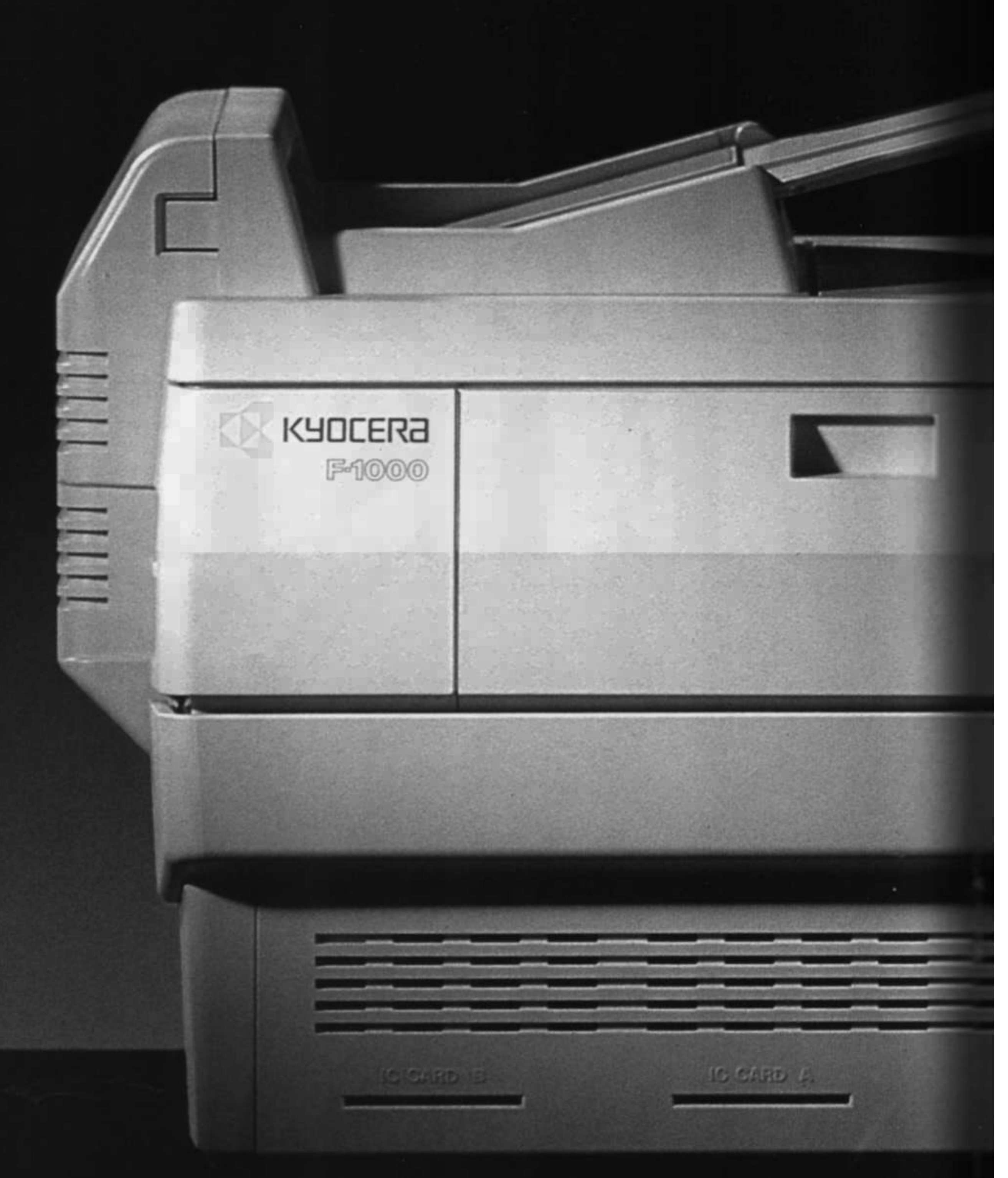
Da das Programm keine speziellen Basic-Befehle verwendet, dürfte es in allen Basic-Dialekten lauffähig sein.

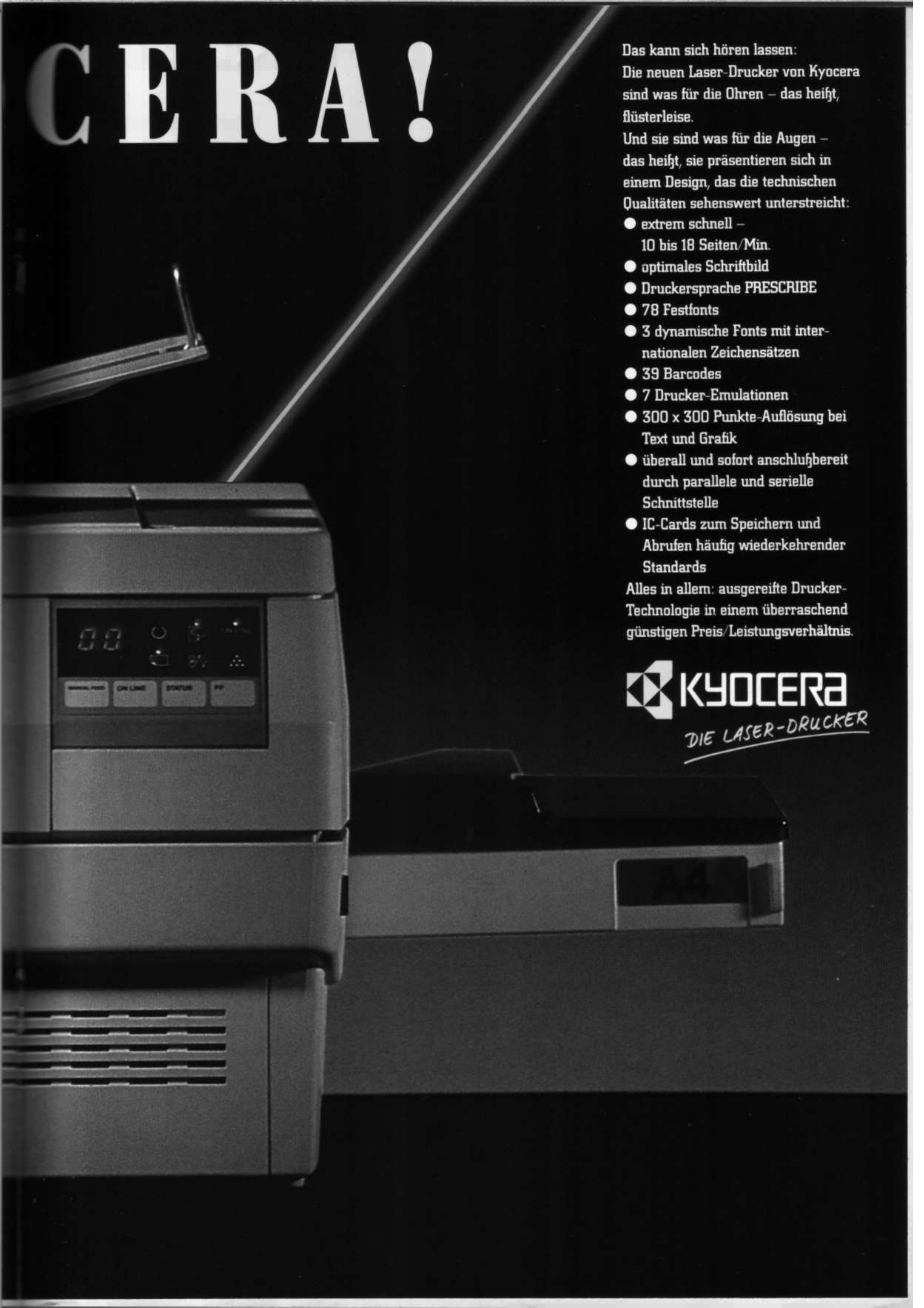
Wolfgang Bethmann

```
10 REM ************************************************************
20 REM *        Rentenberechnungsprogramm 1942 - 1986             *
30 REM *        W.Bethmann August 1987                            *
40 REM ************************************************************
50 CLS
60 PRINT "           Rentenberechnung von 1942 - 1986
70 DIM FE(100),BE(100),B(100)
80 FOR K = 42 TO 86
90 READ FE(K)
100 NEXT
110 DATA 2310,2324,2292
120  DATA 1778,1778,1833,2219,2838,3161,3579,3852,4061,4234
130  DATA 4548,4844,5043,5330,5602,6101,6723,7328,7775,8467
140  DATA 9229,9893,10219,10842,11839,13343,14931,16335,18295
150  DATA 20381,21808,23335,24945,26242,27685,29485,30900
160  DATA 32198,33293,34292,35286,35286
170 PRINT:PRINT
180 INPUT"Name und Geburtstag ";N$
190 PRINT "Alle Zeiten als Dezimalbruch eingeben (z.B. 12.75 Jahre)":PRINT
200 INPUT"Für wieviele Versicherungsjahre ";VJ
210 INPUT"Für wieviele Jahre berechnen ";BJ
220 INPUT"1.Versicherungsjahr 19";A:A1=A
230 INPUT"Ab welchem Monat (1 - 12) ";MO:A=A-1+((MO-1)/12)
240 INPUT"Letztes Versicherungsjahr 19";E
245 IF E>86 THEN PRINT E"ist falsch (zu gross)":GOTO 240
250 INPUT"Ersatzzeiten vor 1965 ";AY
260 INPUT"Ersatzzeiten ab 1965  ";AZ
270 INPUT"Allgem.-Rentenbem.-Grundlage (1986 = DM 27885.--)";BA
280 B(I)=0:S=0
285 PRINT:PRINT"Eingabe des jaehrlichen Bruttoverdienstes."
290 FOR I = A1 TO E
300 PRINT "Betrag 19"I"=";:INPUT BE(I)
310 B(I)=BE(I)/FE(I)
320 S=S+B(I):NEXT
330 S1=0:IF A1=>64 GOTO 350
340 FOR I = A1 TO 64:B(I)=BE(I)/FE(I):S1=S1+B(I):NEXT
350 REM
360 BZ=S1/(64-A-AY):IF A1=>64 THEN BZ=1
370 PRINT "Bemess.-Faktor für Ersatz- u.Ausfallzeit";INT(BZ*100)
380 BP=(S+BZ*AY+AZ)/VJ
390 PRINT "Pers.-Bemess.-Faktor in %";INT(BP*100)
400 PRINT "Rentenbemessungsbetrag für 1986  DM";BA
410 RE=BP*BA*BJ*1.5/1200
420 PRINT "Monatsrente ca.DM ";INT(RE)
430 PRINT:PRINT:PRINT
440 INPUT"Ausdrucken (J/N)";Z$
450 IF Z$<>"J" THEN 200
460 LPRINT "                 Rentenberechnung und Entwicklung"
470 LPRINT:LPRINT
480 LPRINT"                 ";N$:LPRINT
490 LPRINT TAB(10)"Jahr";TAB(20)"Verdienst";TAB(30)"Tabellenwert";TAB(51)"%-Wert
"
500 LPRINT
510 FOR I =A1 TO E
520 LPRINT TAB(10)I;TAB(20)BE(I);TAB(30)FE(I);TAB(50)B(I)*100
530 NEXT:LPRINT
540 LPRINT "Versicherungsdauer in Jahren              ";BJ
550 LPRINT "Ausfallzeiten vor 1965 (in Jahren)        ";AY
560 LPRINT "Ausfallzeiten ab  1965 (in Jahren)        ";AZ
570 LPRINT "Rentenbemessungsgrundbetrag  DM           ";BA
580 LPRINT "Faktor für Ausbild.-u.Ersatzzeiten (%)  ";BZ*100
590 LPRINT "Persönlicher Bemessungs-Faktor      (%)  ";BP*100
600 LPRINT
610 LPRINT "Monatliche Altersrente       DM          ";INT(RE)
620 LPRINT "Monatliche Berufsunf.-Rente  DM          ";INT(RE*2/3)
630 GOTO 180
```

Bild. Das kleine GW-Basic-Programm berechnet Ihre Rente

LIEBER KYO




CERA!
Das kann sich hören lassen:
Die neuen Laser-Drucker von Kyocera sind was für die Ohren – das heißt, flüsterleise.
Und sie sind was für die Augen – das heißt, sie präsentieren sich in einem Design, das die technischen Qualitäten sehenswert unterstreicht:
● extrem schnell – 10 bis 18 Seiten/Min.
● optimales Schriftbild
● Druckersprache PRESCRIBE
● 78 Festfonts
● 3 dynamische Fonts mit internationalen Zeichensätzen
● 39 Barcodes
● 7 Drucker-Emulationen
● 300 x 300 Punkte-Auflösung bei Text und Grafik
● überall und sofort anschlußbereit durch parallele und serielle Schnittstelle
● IC-Cards zum Speichern und Abrufen häufig wiederkehrender Standards
Alles in allem: ausgereifte Drucker-Technologie in einem überraschend günstigen Preis/Leistungsverhältnis.
KYOCERA
DIE LASER-DRUCKER
ON LINE
STATUS
A4

DAS KYOCERA LASER-QUARTETT.



Der Laser-Drucker F-1000.
Er ist der kompakte Lowcost-Drucker mit konstant zuverlässiger Leistung. Ideal für die Textbe- und -verarbeitung. In optimaler Qualität bringt er jedes Schriftstück zu Papier. Bei 10 DIN A4-Seiten/Min. Entweder aus der Papierkassette (250 Blatt) oder über den Einzelblatteinzug.

Weitere Leistungsnachweise
78 Festfonts in allen Emulationen
RAM: 512 KByte
RAM-Erweiterung: 1.0 MByte
Video RAM: 512 KByte



Der Laser-Drucker F-1200.
Er ist der starke Drucker für spezielle Anwendungen und Grafik-Lösungen insbesondere im CAD/CAM-Bereich. Denn hier zeigt er sich bestens in Form – schwarz auf weiß, pieksauber Blatt für Blatt. Seine Geschwindigkeit: 10 DIN A4-Seiten/Min. – über die 250-Blatt-Papierkassette oder über den Einzelblatteinzug.

Weitere Leistungsnachweise
78 Festfonts in allen Emulationen
RAM: 1.5 MByte
RAM-Erweiterung: 2.0 MByte
Video RAM: 2.0 MByte



Der Laser-Drucker F-3000.
Er ist der absolute Hochleistungs-Drucker für alle, denen es nicht schnell genug gehen kann. Mit 18 DIN A4-Seiten/Min. erreicht er eine Geschwindigkeit, die für Masse ausgelegt ist – ohne dabei die Klasse, also gestochen scharfe Wiedergabe, zu vergessen. Dabei helfen ihm, wie beim F-2200, das Papier-Doppelschacht-system und der integrierte 5-Fach-Sorter.

Weitere Leistungsnachweise
78 Festfonts in allen Emulationen
RAM: 1.5 MByte
RAM-Erweiterung: 2.0 MByte
Video RAM: 2.0 MByte

Coupon
☐ Bitte schicken Sie mir ausführliche Informationen zum neuen Laser-Drucker
☐ F-1000
☐ F-1200
☐ F-2200
☐ F-3000

Firmenstempel:

Gesprächspartner:

Tel.:

3/a b c d e



Der Laser-Drucker F-2200.
Er ist der Universal-Drucker mit etwas mehr drauf und mehr dran als der F-1200. Und das hilft besonders, wenn große Mengen verarbeitet werden müssen wie z.B. im Textbereich. Deshalb also hat er zwei Papierkassetten für 2 mal 250 Blatt – B5 bis A4 – ebenso wie einen standardmäßig eingebauten 5-Fach-Sorter. Seine Druckgeschwindigkeit: 10 DIN A4-Seiten/Min.

Weitere Leistungsnachweise
78 Festfonts in allen Emulationen
RAM: 1.5 MByte
RAM-Erweiterung: 2.0 MByte
Video RAM: 2.0 MByte

KYOCERA

KYOCERA ELECTRONICS EUROPE GmbH
Emanuel-Leutze-Straße 1B
4000 Düsseldorf 11
Tel. 0211/52980
Telefax 0211/529852
Telex 8588143 KEED



Include-Dateien übersichtlich dargestellt

Entwickelt man ein umfangreiches PASCAL-Programm, so bindet man gerne der Übersicht wegen seine ausgetesteten Procedures oder Functions per Include-Datei ein. Hat die Zahl der Include-Dateien einen gewissen Umfang erreicht, verliert man leicht die Übersicht darüber, welche Procedures mit welchem Namen und welchen Parametern in welcher Include-Datei stehen. Hier schafft das Turbo-Pascal-Programm Inctree Abhilfe (*Bild 1*). Beantwortet man dessen Frage nach der Datei mit INCTREE, dann erzeugt es die in *Bild 2* dargestellte Text-Datei mit dem Namen INCTREE.TTT.

Dieser Text gibt eine komprimierte Übersicht über die benutzten Include-Files mit allen Procedure-Köpfen, wobei alle in Include-Dateien auftretende Procedures durch Einrücken gekennzeichnet sind. Der Text kann editiert und bei Bedarf ausgedruckt werden. Ein solcher Ausdruck ist bei der Weiterentwicklung des Programms sehr nützlich.

Wendet man das Programm IncTree auf die Text-Datei INCTREE.PAS selbst an, so werden in der Function Druck alle Textzeilen, die ein Schlüsselwort wie „Procedure" usw. enthalten, mit ausgedruckt. Diese Zeilen wurden mit einem Editor entfernt.

Wer noch andere Schreibweisen für Procedure und Function verwendet, muß die Function Druck entsprechend erweitern. Das vorliegende Programm hat sich beim Autor in der Entwicklung großer Projekte bewährt.

Dr. Karl-Heinz Krüger

```
{--- DOSDATE.INC ------------------------------------------------}

 TYPE  typdate    = RECORD jahr,mon,tag,wtag : BYTE; END;
 VAR   dosdate    : typdate;

 PROCEDURE GetDate;
  VAR   reg  : RECORD ax,bx,cx,dx,bp,si,di,ds,es,fl : INTEGER END;
        date : typdate;
  BEGIN
   reg.ax:=$2A00; MSDOS(reg);
   WITH dosdate DO
    WITH reg DO
     BEGIN
      wtag:=LO(ax) MOD 7;
      jahr:=cx-1900;
      mon :=HI(dx);  tag:=LO(dx);
     END;
  END;  { GetDate }

 FUNCTION  DateStr (VAR date : typdate) : typstr14;
  CONST wochentag = 'SoMoDiMiDoFrSa      ';
  VAR   st : typstr14;  s : STRING[2];
  BEGIN
   WITH date DO
    BEGIN
     STR(jahr,st);
     STR(mon:2,s);     st:=s+'.'+st;
     STR(tag:2,s);     st:=s+'.'+st;
    END;
   DateStr := COPY(wochentag,1+2*date.wtag,2)+' '+st;
  END;  { DateStr }

{--- CLEAR.INC -------------------------------------------------}

 PROCEDURE ClearDatName (VAR  datei : typstr14);
  VAR  p  : BYTE;
  BEGIN
   WRITE(lc,lc,datei,'    ');
   p:=1;
   REPEAT
    datei[p]:=UPCASE(datei[p]);
    IF datei[p] IN ['0'..'9','A'..'Z','.',':','_','-']
     THEN p:=p+1
     ELSE DELETE(datei,p,1);
   UNTIL p>LENGTH(datei);
   WRITELN(datei);
  END;
 {--- INCTREE.PAS ------------------ Apr. 87 ------------------}

 PROGRAM IncTree;

  CONST
    cr = #13;  lc = #13#10;    rand = '    ';

  TYPE
    typstr14      = STRING [14];


  VAR
    quelfil,incfil,zielfil        :  TEXT;
    queldat,incdat,zieldat        :  typstr14;
    zeile                         :  STRING [125];
    p,q                           :  BYTE;
    rd                            :  STRING [22];
```

```
 {$I DOSDATE.INC }

 FUNCTION  Druck : BOOLEAN;
  BEGIN
   Druck:=(POS('PROGRAM'  ,zeile)>0) OR (POS('program'  ,zeile)>0)
       OR (POS('PROCEDURE',zeile)>0) OR (POS('procedure',zeile)>0)
       OR (POS('FUNCTION' ,zeile)>0) OR (POS('function' ,zeile)>0)
       OR (POS('END.',zeile)>0) OR (POS('end.',zeile)>0);
  END;

 FUNCTION  Include : BOOLEAN;
  VAR  p : BYTE;
  BEGIN
   p:=POS('{$I',zeile);
   IF p=0 THEN p:=POS('{$i',zeile);
   IF p=0 THEN p:=POS('(*$I',zeile);
   IF p=0 THEN p:=POS('(*$i',zeile);
   Include:=(p>0) AND (zeile[p-1]<>#39);     { #39 = ' }
  END;

 PROCEDURE Print;
  BEGIN
   WRITE(lc,rd,COPY(zeile,1,79-LENGTH(rd)));
   WRITE(zielfil,lc,rd,zeile);
  END;

 {$I CLEAR.INC }

 BEGIN
  rd:=rand;
  GetDate;
  WRITE(lc,lc,'Quell-Datei : ');  READLN(queldat);
  ClearDatName(queldat);
  p:=POS('.',queldat);
  IF p=0 THEN queldat:=queldat+'.PAS';
  ASSIGN(quelfil,queldat);     RESET(quelfil);
  p:=POS('.',queldat)-1;
  IF p=0 THEN p:=8;
  zieldat:=COPY(queldat,1,p)+'.TTT';
  ASSIGN(zielfil,zieldat);     REWRITE(zielfil);
  WRITELN(zielfil,lc,rd,queldat,DateStr(dosdate):20,zieldat:20,lc,lc);
  WRITELN(lc,rd,queldat,DateStr(dosdate):20,zieldat:20,lc,lc);
  REPEAT
   rd:=rand;
   READLN(quelfil,zeile);
   IF Druck THEN Print;
   IF Include THEN
    BEGIN
     WRITELN(zielfil,lc,lc,rd,zeile);
     WRITELN(lc,lc,rd,zeile);
     rd:=rand+rand;
     p:=POS('$',zeile)+2;
     incdat:=COPY(zeile,p,14);
     ClearDatName(incdat);
     ASSIGN(incfil,incdat);     RESET(incfil);
     REPEAT
      rd:=rand+rand+rand;
      READLN(incfil,zeile);
      IF Druck THEN Print;
     UNTIL EOF(incfil);
     CLOSE(incfil);
     WRITELN(zielfil);
     WRITELN;
    END;
  UNTIL EOF(quelfil);
  CLOSE(quelfil);
  CLOSE(zielfil);
  WRITELN(lc,zieldat);
 END.
```

Bild 1. Das Programm INCTREE mit zwei Include-Dateien

```
INCTREE.PAS        Fr 11.12.87        INCTREE.TTT

 PROGRAM IncTree;

  {$I DOSDATE.INC }

      PROCEDURE GetDate;                               { DOS }
      FUNCTION DateStr (date : typdate) : typdatestr;   { ww jj.mm.tt  hh:mm }

  FUNCTION  Druck   : BOOLEAN;
  FUNCTION  Include : BOOLEAN;
  PROCEDURE Print;

  {$I CLEAR.INC }

      PROCEDURE ClearDatName (VAR  datei : typdatei);

 END.
```

mc-Redaktion

8000 München 37

Postfach 37 01 20

Bild 2. Hier wurde INCTREE auf sich selbst angewendet

Gerd Graf

Der mc-modular-AT

Teil 6: Der Festplatten-/Floppy-Controller

Wesentliches Element des mc-modular-ATs sind der Festplatten-/Floppy-Controller und das ins BIOS integrierte Setup-Programm. Da immer mehr Rechner der AT-Klasse die gleichen Eigenschaften wie der mc-modular-AT haben, helfen diese Informationen auch Anwendern anderer ATs weiter.

Bei Festplatten- und Floppy-Controllern sollten alle Aufrufe nur über das BIOS geschehen, und auch nur dann, wenn man sich darüber im Klaren ist, was man tut und vorher wichtige Programme und Dateien gesichert hat. Wir beschreiben den Festplatten-/Floppy-Controllers nicht so im Detail wie die Hercules-Karte, da es für den Anwender nicht sinnvoll ist, direkt in die Register des Controllers zu schreiben. Tut man dies dennoch, kann man sehr viel Unheil anrichten. Sogar die Festplatte könnte beschädigt werden.

Bei jedem Rechner mit Festplatte sollte man regelmäßig Sicherungskopien herstellen. Auch der beste Anwender löscht einmal versehentlich Dateien. Wenn man dies nicht bemerkt und anschließend Dateien auf die Festplatte kopiert, helfen auch die berühmten Norton Utilities nicht mehr weiter. Letztendlich kommt es auch bei der besten Festplatte vor, daß Daten in einem „schwarzen Loch" verschwinden. Die Festplattenhersteller empfehlen als Vorbeugemaßnahme, die Platte ab und zu erneut zu formatieren. Zum Erstellen von Sicherungskopien bieten sich verschiedene Medien an. Am billigsten, aber auch am langsamsten, ist der Backup mit Disketten. Professionell ist der Backup mit einem Streamer, der die Daten auf einem speziellen Magnetband sichert, das je nach Bauart in einer Kassette oder einer Cartridge untergebracht ist.

Da die Preise für die Festplattenlaufwerke mittlerweile sehr stark gesunken sind, setzen viele Anwender als Backup-Medium eine zweite Festplatte ein. Sofern kein schwerwiegender Fehler im System auftritt, ist diese Sicherungmethode fast so gut wie die mit Disketten oder Bändern. Vor mechanischen Beschädigungen ist natürlich eine eingebaute zweite Festplatte nicht geschützt, so daß wichtige Daten doch mit einem Streamer oder mit Disketten gesichert werden sollten.

Register des Festplatten-/Floppy-Controllers

Der Festplatten-/Floppy-Controller des mc-modular-ATs ist mit hochintegrierten Bausteinen des Herstellers Western-Digital bestückt. Die Daten zur Programmierung des Controllers werden über zwei Registersätze (*Tabelle*) übergeben:

- zum Festplattenteil des Controllers gehören die Ein-/Ausgabe-Adressen 1F0H...1F7H
- zum Floppyteil des Controllers gehören die E/A-Adressen 3F2H und 3F4H...3F7H

In der Tabelle ist auch von zwei verschiedenen Adressbereichen, dem sogenannten Primär- und dem Sekundär-Adreßbereich die Rede.

Den Festplatten-Floppy-Controller kann man auch auf einen zweiten Speicherbereich (Secondary) einstellen; dieser Speicherbereich liegt für die Festplatte bei den Adressen 170H...177H sowie für die Floppy im Bereich 372H und 374H...377H.

Einige auf dem Markt befindliche BIOS-Versionen können zwei getrennte Festplatten-/Floppy-Controller im System ansprechen.

Dies ist jedoch mit dem Standard-IBM-BIOS des ATs als auch mit den uns bekannten BIOS-Versionen (Award sowie Phoenix) nicht möglich. Die Secondary-Adressen sind nur der Vollständigkeit halber erwähnt.

Die Brücken zum Umschalten auf den zweiten Adreßbereich (P3 und P5) darf man nicht verändern, da sich sonst die Festplatte oder die Floppy tot stellt.

Ohne Brücken läuft nichts

In *Bild 1* sind die Lagen der Brücken und die Verbindungsleisten auf dem Festplatten-/Floppy-Controller dargestellt.

Man erkennt, daß zum Anschluß der Floppy nur eine Verbindungsleiste (J5) notwendig ist; zum Anschluß einer Festplatte benötigt man zwei Verbindungsleisten und damit auch zwei Kabel (J3, J4). Auf die Pfostenleiste J4 wird das Flachbandkabel mit den Steuerleitungen (34polig) gesteckt, das bereits für den Anschluß von zwei Festplatten vorbereitet ist. Das Flachbandkabel mit den Datenleitungen für die erste Festplatte wird auf die Pfostenleiste J3 gesteckt. Zum Anschluß der Datenleitungen einer zweiten Festplatte dient die Pfostenleiste J2. Die möglichen Stellungen der Brücken sind in *Bild 2* zu sehen.

Nach unseren Erfahrungen kann die Brücke P8, die festlegt, ob eine oder zwei Festplatten im System sind, bereits beim Einsatz nur einer Festplatte auf zwei Festplatten eingestellt werden.

Tabelle: Belegung der E/A-Adressen des Festplatten-/Floppy-Controllers

Primär	Sekundär	Lesen	Schreiben
1F0	170	Daten-Register	Daten-Register
1F1	171	Fehler-Register	Schreib-Prekompensation
1F2	172	Sektor-Zähler	Sektor-Zähler
1F3	173	Sektor-Nummer	Sektor-Nummer
1F4	174	Zylinder Low-Byte	Zylinder Low-Byte
1F5	175	Zylinder High-Byte	Zylinder High-Byte
1F6	176	Laufwerk/Kopf	Laufwerk/Kopf
1F7	177	Status-Register	Kommando-Register
3F2	372	–	Ausgangs-Register
3F4	374	Haupt-Status-Reg.	Haupt-Status-Reg.
3F5	375	Floppy-Daten-Reg.	Floppy-Daten-Reg.
3F6	376	–	Festplattenregister
3F7	377	Eingangs-Register	Floppy-Steuer-Register

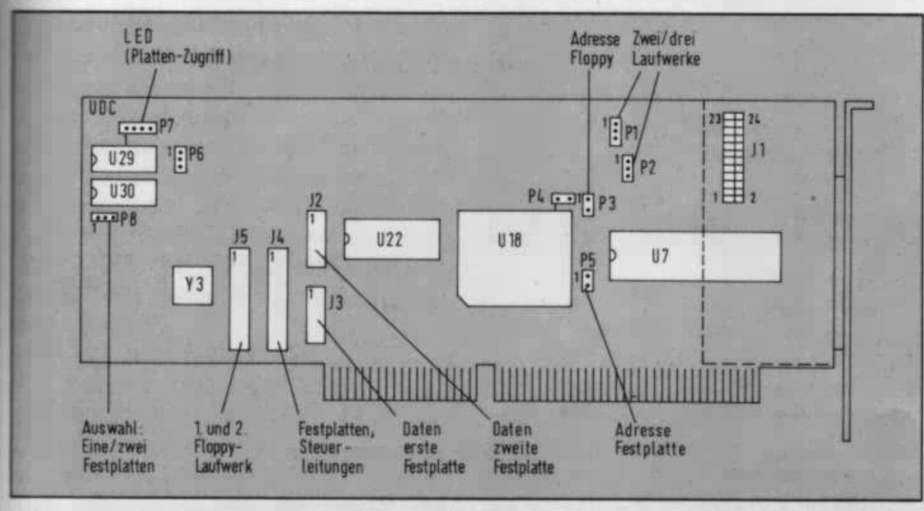


Bild 1. Lage der Brücken und Verbindungsleisten

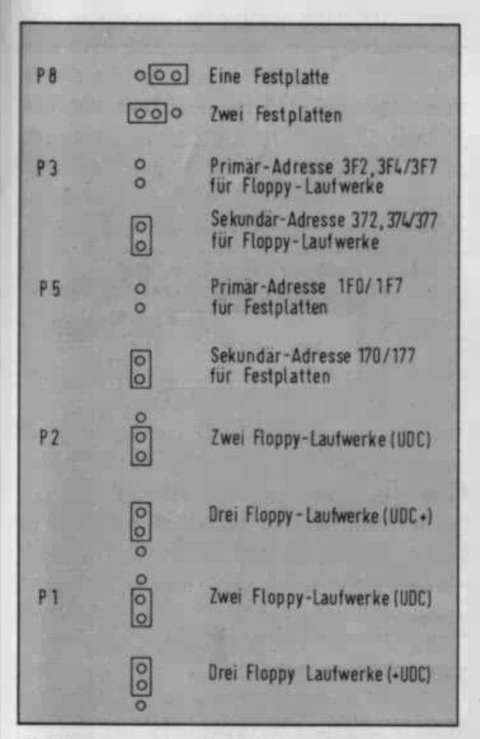


Bild 2. So werden die Brücken richtig gesetzt

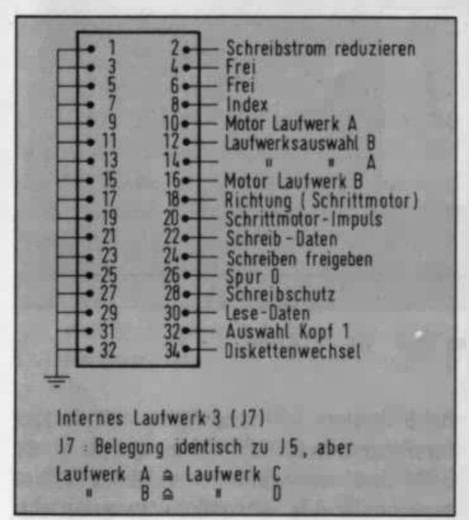


Bild 3. Der Floppy-Anschluß

In *Bild 3* ist die Floppy-Schnittstelle dargestellt – im Bestückungsplan ist sie als J5 gekennzeichnet.

Drei Laufwerke auf einem Streich

Der Festplatten-/Floppy-Controller steuert maximal zwei Diskettenlaufwerke sowie zwei Festplatten. Mit der Huckepack-Baugruppe UDC plus kann er ein drittes Diskettenlaufwerk ansteuern. Auf der Erweiterungsplatine befindet sich ein Anschluß für ein drittes internes Laufwerk (J7) sowie ein Anschluß für ein externes drittes Laufwerk, das später beschrieben wird.

Der Anschluß für das interne dritte Laufwerk ist identisch zu den Anschlüssen für die Laufwerke 1 und 2. Lediglich die Bezeichnungen Laufwerk A und Laufwerk B sind hier durch Laufwerk C bzw. Laufwerk D zu ersetzen.

Wichtig ist, daß der mc-modular-AT gemischt mit 3½-Zoll- und 5¼-Zoll-Laufwerken bestückt werden darf. Für jedes dieser Laufwerke sind mit dem Setup-Programm vier Speicherkapazitäten einsteldbar: 360 KByte, 720 KByte, 1,2 MByte und 1,44 MByte.

Vom Controller werden 5¼-Laufwerke für Disketten mit 360 KByte und 1,2 MByte unterstützt und 3½-Laufwerke für Disketten mit 720 KByte und 1,44 MByte (IBM-PS/2-kompatibel). Um auf einer 3½-Diskette 1,44 MByte zu speichern, benötigt man ein hochwertiges Laufwerk, wie z. B. das Modell FD 35 H-FN von TEAC. Auf den Kennbuchstaben H kommt es an; das Laufwerk G-FN kann nur 1,2 MByte auf einer Diskette speichern.

Die Festplattensteuersignale (*Bild 4*) werden an beide Festplatten (sofern vorhanden) geführt. Über ein teilweise verdrehtes

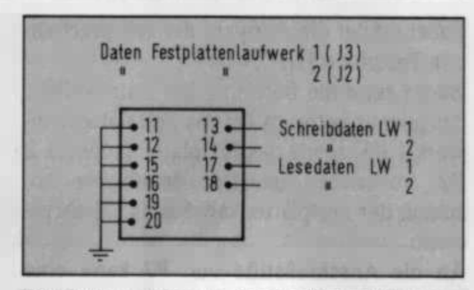


Bild 5. Anschlußbelegung der Datenleitungen

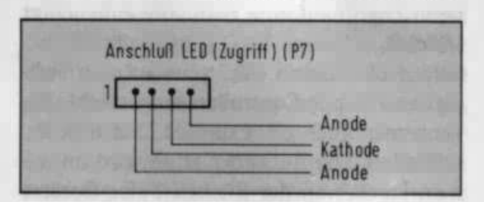


Bild 6. Anschluß der Zugriffs-LED

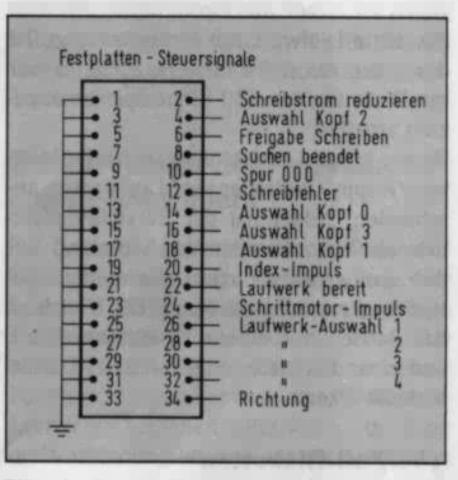


Bild 4. Anschlußbelegung des Festplatten-Anschlusses

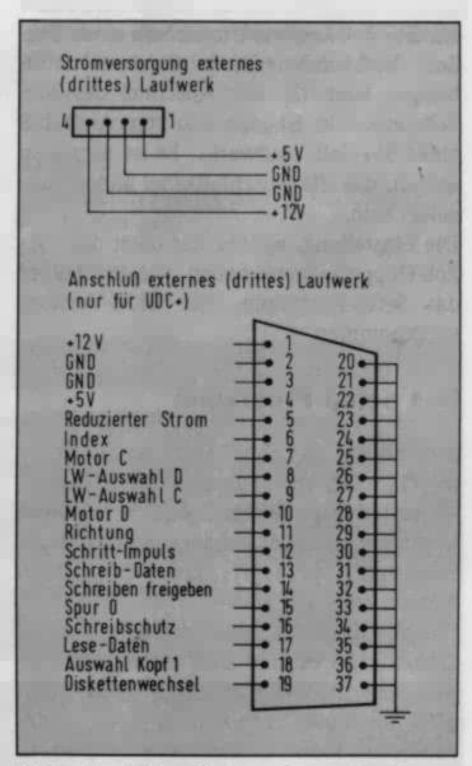


Bild 7. Anschlüsse für ein externes Laufwerk

Kabel erfolgt die Auswahl der entsprechenden Festplatte [1].

Bild 5 zeigt die Belegung des Datenkabels, das jeweils getrennt für das Festplattenlaufwerk 1 (J3) sowie das Festplattenlaufwerk 2 (J2) vorhanden ist. Der detaillierte Anschluß der Festplattenkabel ist in [1] dargestellt.

An die Anschlußstifte von P7 kann eine Leuchtdiode angeschlossen werden, die beim Zugriff auf eine Festplatte aufleuchtet (*Bild 6*).

Sofern die Platine UDC plus auf den Festplatten-/Floppy-Controller aufgesteckt ist, kann man auch ein externes Laufwerk anschließen. Die Aufsteckplatine wird mit einem Stecker an der Rückseite des Gerätes verbunden, dessen Belegung in *Bild 7* angegeben ist. Über die Pfostenleiste P9 wird das dritte Laufwerk mit Strom versorgt. Ist das dritte ein 3½-Laufwerk, kann es nur mit Disketten von 720 KByte Speicherkapazität arbeiten.

Bis vor kurzem konnte man an den Festplatten-/Floppy-Controller vier Laufwerken anschließen. Leider hat der Hersteller plötzlich ohne Kommentar den Anschluß auf drei Laufwerke reduziert. Vermutlich bestanden Probleme mit dem BIOS. Üblich ist der Betrieb mit einem Diskettenlaufwerk und einer Festplatte oder mit zwei Diskettenlaufwerken.

3½-Zoll-Disketten

Der Hersteller des mc-modular-ATs bietet einen Einbausatz an, damit der Anwender ein 3½-Zoll-Laufwerk innerhalb eines 5¼-Zoll- Einbaurahmens im Gehäuse unterbringen kann [2]. Der Anschluß des 3½-Zoll-Laufwerks ist identisch zum Anschluß eines 5¼-Zoll- Laufwerks. Es ist darauf zu achten, daß die Anschlußkabel anders ausgelegt sind.

Die Einstellung, welche Kapazität das 3½-Zoll-Floppylaufwerk haben soll, wird über das Setup-Programm (nur BIOS-Version) vorgenommen.

Das Setup-Programm

Der IBM-AT und damit auch der mc-modular-AT speichert systemspezifische Daten im batteriegepufferten CMOS-RAM. Diese Daten können über ein sogenanntes Setup-Programm verändert bzw. eingetragen werden.

Früher gab es das Setup-Programm nur auf Diskette. Da es eine allzu menschliche Eigenschaft ist, wichtige Dinge gerade dann nicht zu finden, wenn man sie am dringendsten braucht, haben verschiedene BIOS-Hersteller das Setup-Programm im

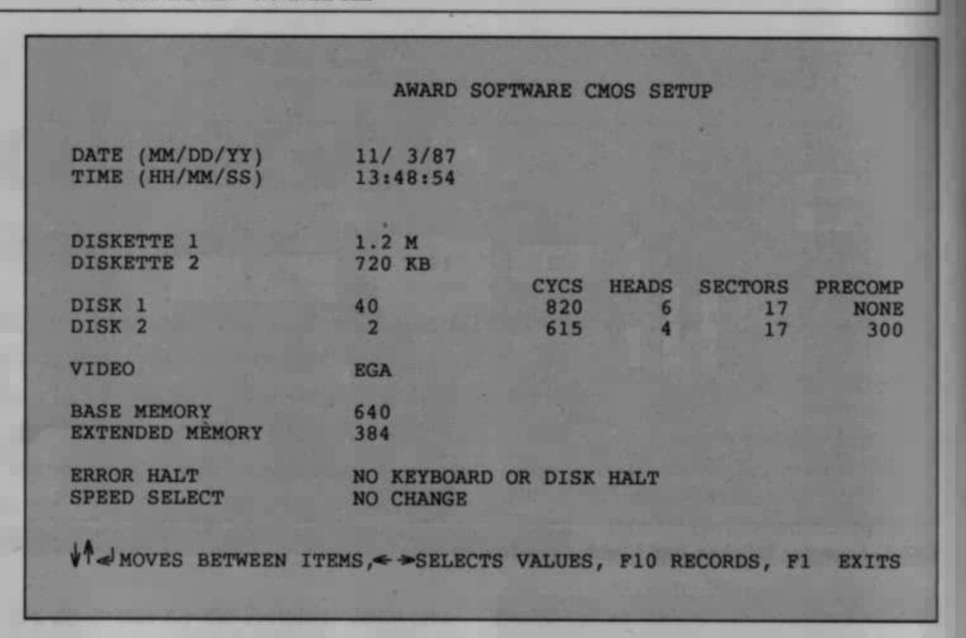


Bild 8. Das Menü des Setup-Programms

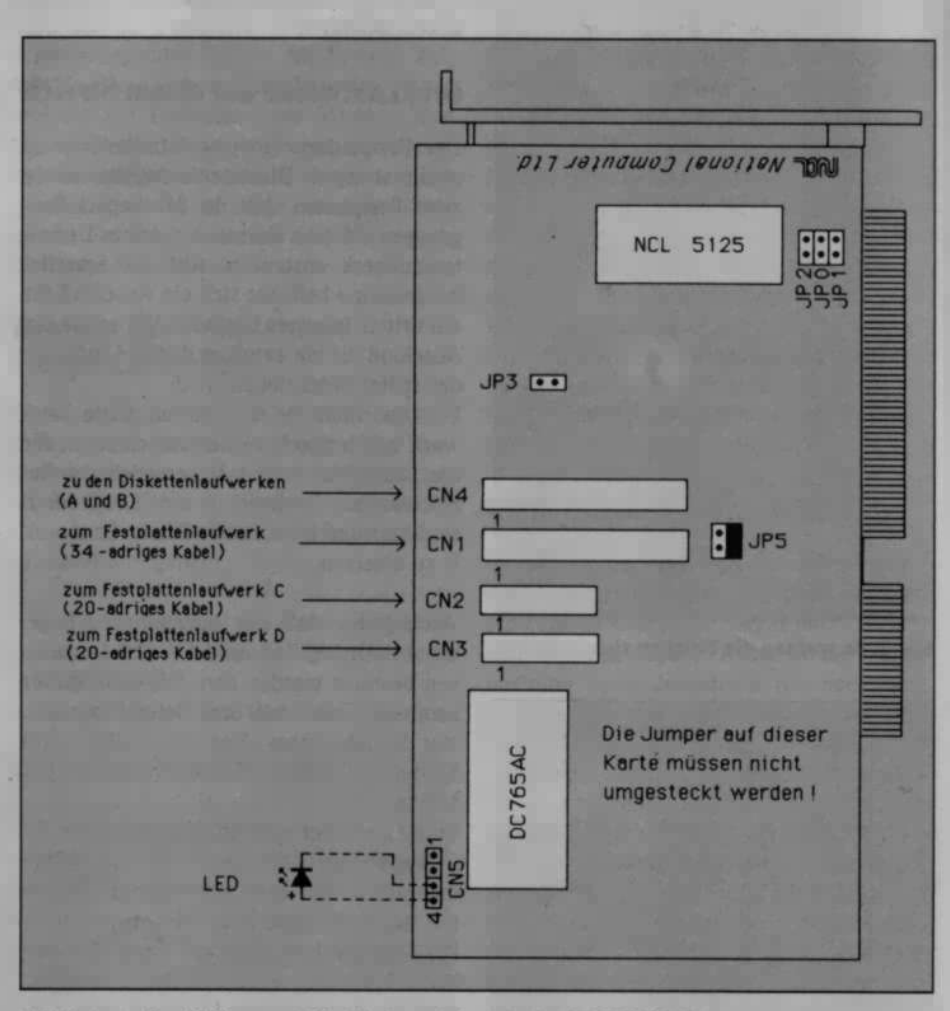


Bild 9. Lage der Brücken und Verbindungsleisten beim NDC 5125-80

BIOS integriert. Die ersten Versionen des mc-modular-ATs wurden noch mit einer Setup-Diskette ausgeliefert.

Bei den ab ca. November ausgelieferten BIOS-Versionen des mc-modular-ATs (ab BIOS-Version 3.01 und höher) ist das Setup-Programm im EPROM integriert. Ältere BIOS-Versionen werden vom Hersteller des mc-modular-AT kostenlos umgetauscht. Diese BIOS-Versionen werden auch benö-

tigt, falls die 3½-Zoll-Diskettenlaufwerke mit der Speicherkapazität von 720 KByte oder 1,44 MByte (IBM PS/2 kompatibel) betrieben werden sollen.
Um in das integrierte Setup-Menü zu kommen, drückt man beim Booten des Systems irgendeine Taste; damit wird dem Tastatur-Prüfprogramm ein Fehler gemeldet und das AWARD-BIOS schreibt auf den Bildschirm den Hinweis „Setup mit Control+Shift+Escape aufrufen".
Nach der gleichzeitigen Betätigung dieser drei Tasten erscheint das Setup-Menü gemäß *Bild 8*. Die einzugebenden Felder sind hierbei invers dargestellt.
Mit Hilfe der Cursortasten Pfeil nach oben, Pfeil nach unten und der Eingabetaste, kann man eine der Eingabefelder auswählen. Mit Hilfe der Cursortasten Pfeil links und Pfeil rechts, werden die gewünschten Werte eingestellt.
Mit dem Betätigen der F10-Taste speichert man die so ausgewählten Werte im CMOS-Speicher; mit der Funktionstaste F1 beendet man das Setup-Menü, ohne die Änderungen zu speichern.

Vorsicht beim Eingeben des Festplattentyps

Das Datum und die Zeit wird mit den Zifferntasten eingegeben. Wichtig ist, daß beim Setup-Menü das Datum in amerikanischer Schreibweise (Monat/Tag/Jahr) eingegeben werden muß. Die Zeit kann man jedoch in deutscher Schreibweise (vierundzwanzig Stunden) eingeben. Die Speicherkapazität der Diskettenlaufwerke (Disk 1 und Disk 2) kann man ebenso festlegen. Mit der linken oder rechten Pfeiltaste legt man die Speicherkapazität auf 360 KByte, 720 KByte, 1,2 MByte oder 1,4 MByte fest. Natürlich müssen die verwendeten Laufwerke auch für die gewählte Speicherkapazität geeignet sein.
Mit NONE wird kein Diskettenlaufwerk – z. B. für Disk 2 – eingestellt.
„Disk 1" sowie „Disk 2" sind Festplatten. Im Menü sind die Festplattentypen 1...40 bereits enthalten. Der entsprechende Typ der Festplatte ist normalerweise auf dem Aufkleber Ihrer Festplatte angegeben. Die Festplatte ST225 ist vom Typ 2.
Achten Sie bitte darauf, daß Sie den Festplattentyp nicht mehr ändern, nachdem Sie ihn richtig eingestellt haben. Ein totaler Datenverlust könnte die unangenehme Folge sein.
Unter „Video" gibt man die Art des Video-Controllers an. Standard ist beim mc-modular-AT der Hercules-kompatible monochrome Adapter; wählen Sie also hier monochrom aus.

Der mc-modular-AT auf einen Blick

CPU:	80286, 10 MHz, 1 WS bei 120 ns RAMs
FPU:	Einbauplatz für 80287 vorhanden
Speicher:	512 KByte RAM, erweiterbar auf 1 MByte Mit Intel Above Board PS/286 erweiterbar
BIOS:	AWARD-BIOS mit integriertem Setup-Programm
Floppy:	Zwei bzw. drei Disketten-Laufwerke
Festplatte:	Platz für zwei Festplatten

In den folgenden beiden Zeilen kann man die Aufteilung des Speichers bestimmen. Der in Bild 8 gezeigte Vorschlag weist einen maximal möglichen Arbeitsspeicher von 640KB sowie einen erweiterten Speicher von 384 KByte (Extended Memory) auf, das unter MS-DOS als RAM-Floppy dient.

Fehlermeldungen unterdrücken

Sollte das System nur mit 512 KByte bestückt sein, gibt man unter Base Memory 512KB und unter Extended-Memory 0 an.
Bei der Funktion „ERROR HALT" kann der Anwender auswählen:

- HALT ON ALL ERRORS: Der Computer hält bei jedem Fehler
- NO HALT ON ANY ERRORS: Der Computer ignoriert sämtliche Fehler (gefährlich!)
- NO KEYBOARD ERROR HALT: Tastaturfehler werden unterdrückt
- NO DISK ERROR HALT: Das System ignoriert Laufwerksfehler
- NO KEYBOARD OR DISK HALT: Laufwerksfehler und Tastaturfehler werden unterdrückt

Die einzelnen Funktionen werden mit den Pfeiltasten aufgerufen.
Mit der Funktion „SPEED SELECT" kann man zwischen „LOW, HIGH, NO CHANGE" wählen. Ist die CPU-Baugruppe auf 10-MHz-Takt eingestellt, kann man sie dennoch auf 8-MHz-Takt umstellen, wenn man den Menüpunkt „LOW" auswählt. Falls sie bereits mit der Brücke J7 auf 8-MHz-Takt eingestellt ist, kann man mit der Funktion „SPEED SELECT" nichts mehr ändern.
Das Setup-Menü wird verlassen, indem man die F10-Taste drückt. Will man die darauf folgende Kontrollfrage bejahen, muß man wegen der deutschen Tastatur statt der Y-Taste die Z-Taste drücken. Wenn man die Daten nicht speichern will, muß man die N-Taste drücken.

Zweite Version des Festplatten-/Floppy-Controllers

Bei dem Festplatten-/Floppy-Controller UDC gibt es manchmal Lieferengpässe. Der Lieferant des mc-modular-AT weicht dann auf ein zweites Produkt aus, den Controller NDC 5125-80. Er ist identisch in seiner Funktion, kann aber nur zwei Disketten-Laufwerke bedienen. *Bild 9* zeigt die Lage der Verbindungsleisten und der Brücken.

Die Grundversion des mc-modular-ATs ist nun beschrieben. Im nächsten Beitrag schildern wir die Inbetriebnahme des Systems sowie die wichtigsten MS-DOS-Funktionen.

Literatur

[1] Handbuch zum mc-modular-AT. Graf Elektronik, Kempten.
[2] *Graf, Gerd:* Der mc-modular-AT. mc 1987, Heft 9, Seite 36.

Spruch des Monats

„Der Fortschritt der Wissenschaft verändert sich mit der Anzahl der veröffentlichten Zeitschriften."

Parkinson's sechstes Gesetz

Thomas Erbert

Ein Maus-Interface für den mc-68000-Computer

Wer hat sich nicht schon geärgert, wenn er mit einem Joystick am mc-68000-Computer Grafiken bearbeiten wollte: Da wackelt der Bildpunkt, auch wenn man den Joystick nicht bewegt. Mit nur wenig Hard- und Software kann man dieses Problem jedoch sehr einfach und schnell beheben.

Öffnet man eine Maus (z. B. die Atari-Maus), so sieht man vier Lichtschranken, die beim Bewegen der Maus phasenverschobene TTL-Rechteckimpulse erzeugen. Jede dieser Lichtschranken steht dabei für eine Richtung (Links, Rechts, Oben, Unten). Bewegt man die Maus beispielsweise nach rechts, werden nur phasenverschobene Impulse auf den Leitungen „Rechts" und „Links" erzeugt. In dem Phasenversatz steckt die Information für die Bewegungsrichtung. Die Hardware für die Maus muß nur diesen Phasenversatz interpretieren.

Da die Schaltung nicht aufwendig ist, kann sie einfach auf einer Lochrasterplatine in Fädeltechnik aufgebaut und dann in Steckplatz 2 des mc-68000-Computers gesteckt werden. Ist dieser Steckplatz schon belegt, muß man nur das Label „slot" in der Datei MausC.S auf den gewünschten Wert setzen. Die Pin-Bezeichnungen im Schaltplan für den Mausanschluß gelten für eine Atari-Maus. Es kann aber auch jede andere Maus ange-

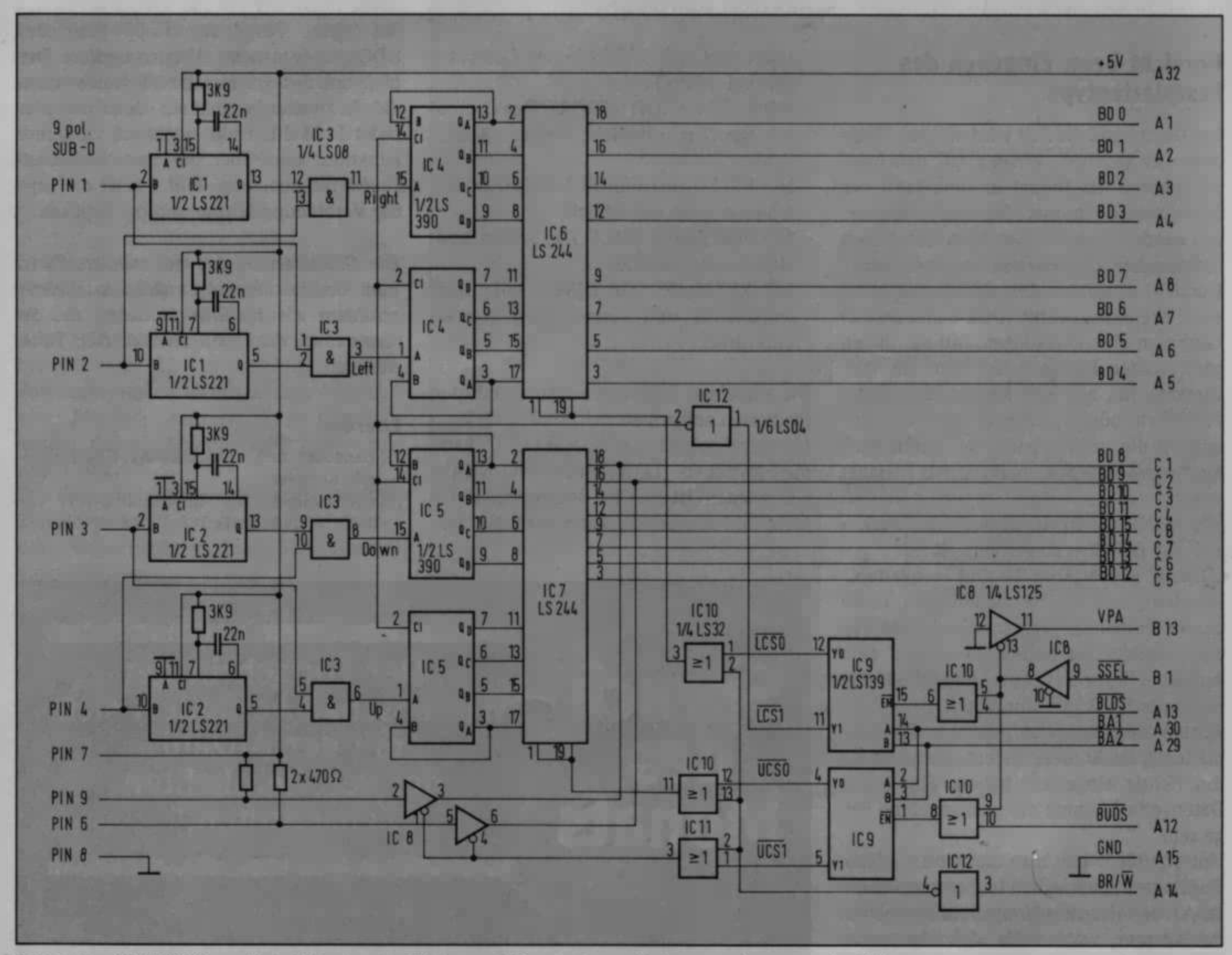


Bild 1. Das IC 74LS32 gewährleistet, das die Zähler- und Maustastenzustände nur gelesen werden können

schlossen werden, die keinen V.24-Ausgang hat (z. B. IBM-Maus).

Die Schaltung in *Bild 1* läßt sich in vier Teile gliedern: Am Eingang erzeugen vier Monoflops (74LS221) aus den ankommenden Rechteckimpulsen, deren Impulsfolgefrequenz von der Bewegungsgeschwindigkeit der Maus abhängt, kurze Impulse von rund 40 µs Dauer. Die Impulsbreite ist dabei unerheblich. Sie darf nur nicht größer als die höchste auftretende Frequenz der ankommenden Rechteckimpulse sein. Die Monoflop-Impulse werden dann mit denen der jeweils entgegengesetzten Richtung UND-verknüpft (74LS08), um nur solche Impulse von den Zählern (74LS390) zählen zu lassen, die zu der Bewegungsrichtung gehören. Der Zählerstand kann über die Bustreiber 74LS244, die Maustasten-Zustände über den Bustreiber 74LS125 auf den Datenbus gelegt werden. Der Adreßdecoder (74LS139) sorgt dabei auch gleich für das Rücksetzen der Zähler.

Bild 2 zeigt das Zeitdiagramm bis zu den Zählereingängen für die zwei Bewegungsrichtungen links und rechts.

Die Software

Bild 3 zeigt das Maschinenprogramm-Modul mit der Initialsierung des Adreßdecoders, den Grafikroutinen und dem Einlesen der Mauswerte. Wichtig ist dabei, daß die Initialsierung des Adreßdekoders nur einmal aufgerufen werden darf.

Die Routine zum Zeichnen einer Linie ist komplett neu geschrieben worden: Die Linien, die durch diese Routine gezeichnet werden, sind wesentlich dünner als die der bisher erschienenen Routinen. Die Dokumentation im Sourcelisting erübrigt eine weitere Erklärung.

Die Übergabewerte der Routine zum Setzen eines Punktes auf dem Grafikbildschirm [1] sind gleich geblieben, so daß diese Zeichenroutine ohne weiteres das alte Programm ersetzen kann. Die Bildschirmbasisadresse wird aus dem Vektor geholt, den die Monitorversion 1.4 zur Verfügung stellt. Ist man in Besitz einer Busterminierungskarte, kann man die Einleseroutine natürlich auch interruptgesteuert über den 300-Hz-Timer starten und nicht, wie im Beispielprogramm, zyklisch in einer Programmschleife. Wird eine interruptgesteuerte Abfrage gewählt, muß man natürlich den Zeiger für den Interrupt-Einsprung auf die Adresse des Labels „_rdcurpos" setzen. Der Ansprung der VOID Funktion „rdcurpos" im C-Programm braucht dann natürlich nicht mehr programmiert zu werden.

Doch nun zu der Beschreibung der Einleseroutine. Die Zähler der Maus-Schnittstelle

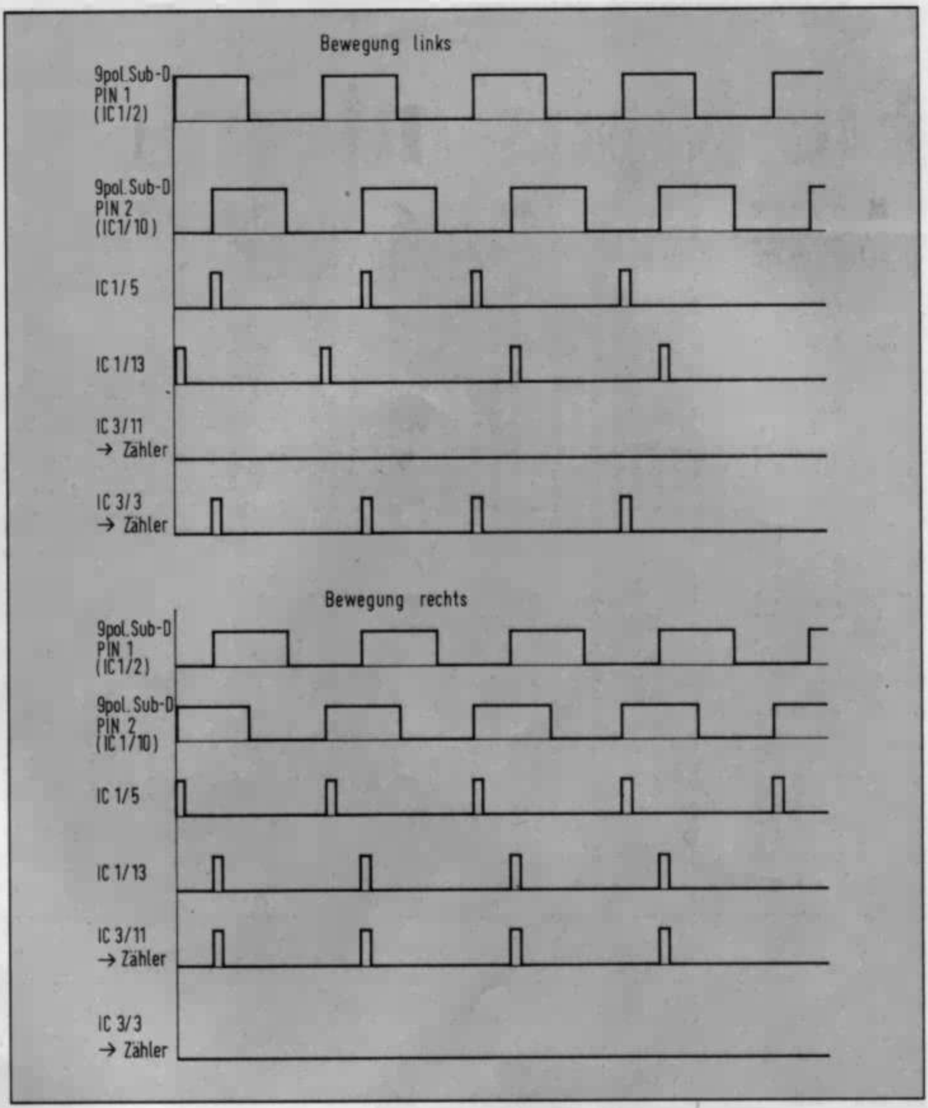
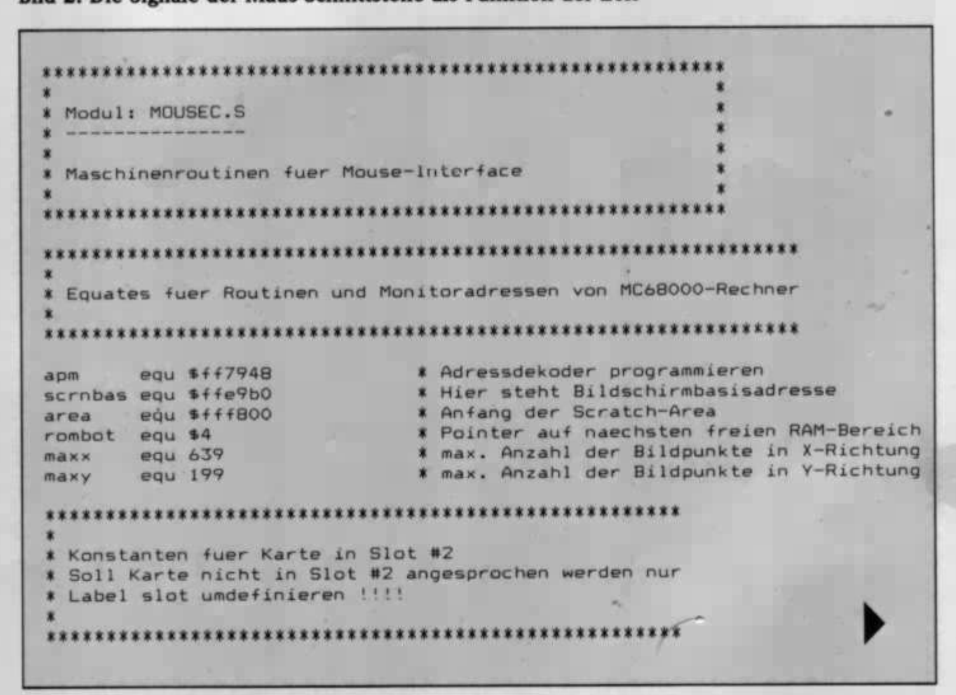


Bild 2. Die Signale der Maus-Schnittstelle als Funktion der Zeit



```
*******************************************************
*                                                     *
* Modul: MOUSEC.S                                     *
* ---------------                                     *
*                                                     *
* Maschinenroutinen fuer Mouse-Interface              *
*                                                     *
*******************************************************

*************************************************************
*
* Equates fuer Routinen und Monitoradressen von MC68000-Rechner
*
*************************************************************

apm     equ $ff7948          * Adressdekoder programmieren
scrnbas equ $ffe9b0          * Hier steht Bildschirmbasisadresse
area    equ $fff800          * Anfang der Scratch-Area
rombot  equ $4               * Pointer auf naechsten freien RAM-Bereich
maxx    equ 639              * max. Anzahl der Bildpunkte in X-Richtung
maxy    equ 199              * max. Anzahl der Bildpunkte in Y-Richtung

***************************************************
*
* Konstanten fuer Karte in Slot #2
* Soll Karte nicht in Slot #2 angesprochen werden nur
* Label slot umdefinieren !!!!
*
***************************************************
```

Bild 3. Per Maschinenprogramm werden Adreßdecoder und Grafikroutinen initialisiert sowie die Zähler eingelesen

```
slot    equ $fa                     * Mousekarte steckt in Slot #2
offs    equ slot-$f8*16             * Offset in der Scratch-Area
kennung equ $0aff0183               * Kennung der Karte in Slot #2
*
* Routinen C-Programm zugaenglich machen
*
        .GLOBL  _drawlin            * Zeichne oder loesche Linie
        .GLOBL  _setpoint           * Setze oder loesche Punkt
        .GLOBL  _cardini            * Initialisiere Karte in Slot #2
        .GLOBL  _rdmouse            * Lese Mousewerte in Speicher ein
        .GLOBL  _kreuz              * Zeichne Kreuz auf Bildschirm
*
* Speicheradressen C-Programm zugaenglich machen
*
        .GLOBL  _deltay             * Bewegung in Y-Richtung
        .GLOBL  _deltax             * Bewegung in X-Richtung
        .GLOBL  _tast0              * Wert von Taste0 von Mouse
        .GLOBL  _tast1              * Wert von Taste1 von Mouse

        .text
        .page

*****************************************************
*                                                   *
* Routine: _drawlin                                 *
* -----------------                                 *
*                                                   *
*    x1,y1                x2,y2                     *
*      +                    +                       *
*                                                   *
* Zeichnen einer Linie von x1,y1 nach x2,y2         *


*                                                   *
* Aufruf von C-Programm aus:                        *
*  int clrflag;      /* Punkt setzen oder loeschen */ *
*  int x1,y1,x2,y2;  /* Koordinaten der Gerade */   *
*  drawlin(x1,y1,x2,y2,clrflag);                    *
*                                                   *
*****************************************************

_drawlin:
        jsr     savreg              * rette alle Register ausser A7
        move.w  #4(sp),d2           * x1-Position holen
        move.w  #6(sp),d3           * y1-Position holen
        move.w  #8(sp),a2           * x2-Position holen
        move.w  #10(sp),a3          * y2-Position holen
        move.w  #12(sp),clrflg      * Flag zum Setzen oder Loeschen holen
        jsr     draw                * zeichne Linie
        jsr     getreg              * hole alle Register wieder ausser A7
        rts                         * zurueck in C-Programm

        .page

*****************************************************
*                                                   *
* Routine: _kreuz                                   *
* ---------------                                   *
*                                                   *
* Setzen oder Loeschen des Kreuzes                  *
*                                                   *
* Aufruf von C-Programm aus:                        *
*  int clrflag;      /* Punkt setzen oder loeschen */ *
*  int x1,y1;        /* Mittelpunkt von Kreuz */    *
*  setpoint(x1,y1,clrflag);                         *
*                                                   *
*****************************************************

_kreuz:
        jsr savreg                  * rette alle Register ausser a7
        move.w #4(sp),x             * hole X-Position
        move.w #6(sp),y             * hole Y-Position
        move.w #8(sp),clrflg        * hole Flag fuer Loeschen oder Zeichnen
*
* Berechne Koordinatenpunkte von horizontaler Linie
*
        move.w x,d2                 * Mittelpunkt von X-Position >= 0 ?
        bge    kreuz1               * ja ->
        clr.w  d2                   * x-Position zu klein
        move.w d2,x                 * x-Position = 0
        movea.w d2,a2               * X-Mittelpunkt nach A2
        addq.w #4,a2                * berechne X-Endposition
        bra    kreuz3
kreuz1:
        cmp.w  #4,d2                * X-Mittelpunkt groesser 4 ?
        bgt    kreuz2               * ja ->
        movea.w d2,a2               * Mittelpunkt auch nach X-Endposition
        addq.w #4,a2                * berechne X-Endposition
        clr.w  d2                   * X-Anfangsposition = 0
        bra    kreuz3
kreuz2:
        movea.w d2,a2               * Mittelpunkt auch nach X-Endposition
        subq.w #4,d2                * berechne X-Startposition
        addq.w #4,a2                * berechne X-Endposition
kreuz3:

        move.w y,d3                 * Mittelpunkt von Y-Position >= 0 ?
        bge    kreuz4               * ja ->
        clr.w  d3                   * sonst Y-Position = 0
        move.w d3,y                 * und ablegen
kreuz4:
        movea.w d3,a3               * Y-Anfangs/Endposition gleich
        jsr draw                    * zeichne oder loesche Linie von Kreuz
*
* berechne Koordinatenpunkte von vertikaler Linie
*
        move.w y,d3                 * Mittelpunkt von Y-Position
        movea.w d3,a3               * Mittelpunkt auch nach Y-Endposition
        addq.w #2,a3                * berechne Y-Endposition
        cmp.w  #2,d3                * wird Y-Anfangsposition zu klein ?
        bge    kreuz5               * nein ->
        clr.w  d3                   * Y-Anfangsposition = 0
        bra    kreuz6
kreuz5:
        subq.w #2,d3                * berechne Y-Startposition
kreuz6:
        move.w x,d2                 * Mittelpunkt von X-Position
        movea.w d2,a2               * auch nach a2
        jsr draw                    * zeichne oder loesche Linie von Kreuz
        jsr    getreg               * hole alle Register wieder ausser A7
        rts                         * zurueck in C-Programm

        .page
```

```
*************************************************************
*                                                           *
* Routine: _setpoint                                        *
* ------------------                                        *
*                                                           *
* Setzen/Invertieren/Loeschen eines Bildschirmpunktes       *
*                                                           *
* Aufruf von C-Programm aus:                                *
*  int clrflag;    /* Punkt setzen/invert./loeschen */      *
*  int x1,y1;      /* Koordinaten der Gerade */             *
*  setpoint(x1,y1,clrflag);                                 *
*                                                           *
*************************************************************

_setpoint:
        jsr     savreg          * rette alle Register ausser A7
        move.w  #4(sp),d2       * x-Position holen
        move.w  #6(sp),d3       * y-Position holen
        move.w  #8(sp),clrflg   * Flag zum Setzen oder Loeschen holen
        jsr     setpnt          * setze oder loesche Punkt
        jsr     getreg          * hole alle Register wieder ausser A7
        rts                     * zurueck in C-Programm

        .page

*************************************************************
*                                                           *
* Routine: _cardini                                         *
* -----------------                                         *
*                                                           *
* Initialisiere Karte in Slot "slot"                        *
* und loesche Zaehlerstaende                                *
*                                                           *
* Aufruf von C-Programm aus:                                *

*  cardini()                                                *
*                                                           *
*************************************************************

_cardini:
        jsr     savreg          * rette alle Register ausser A7
        moveq   #slot,d0        * Slotnummer nach d0
        movea.l rombot,a0       * ROMBOT holen
        subi.l  #$4000,rombot   * ROMBOT korrigieren
        jsr     apm             * Adressdekoder programmieren
        move.l  #kennung,area+offs  * Kennung der Karte eintragen
        move.l  a0,area+4+offs  * Basisadresse merken
        move.w  $2(a0),d0       * loesche Mouse-Zaehler
        jsr     getreg          * hole alle Register wieder ausser A7
        rts                     * zurueck in C-Programm

        .page

*************************************************************
*                                                           *
* Routine: _rdmouse                                         *
* ----------------                                          *
*                                                           *
* Lies Mousewerte in Speicheradressen ein                   *
*                                                           *
* Aufruf von C-Programm aus:                                *
*  rdmouse()                                                *
*                                                           *
*************************************************************

_rdmouse:
        jsr     savreg          * rette alle Register ausser A7
        movea.l area+4+offs,a0  * Lade Basisadresse von Mouse-Karte
        clr.w   d3
        clr.w   d4
        move.w  $0(a0),d1       * hole Mouse-Zaehlerwerte
        move.w  $2(a0),d2       * hole Tastenwerte und loesche Zaehler
        move.b  d1,d3           * horizontale Werte nach d3
        lsr.w   #8,d1           * vertikale Werte nach unten
        move.b  d1,d4           * vertikale Werte nach d4
        lsr.b   #4,d4           * UP-Zaehlerwert in unteres Nibble
        and.w   #$000f,d1       * UP-Zaehlerwert ausblenden
        sub.w   d4,d1           * Delta-Y Wert berechnen ( DOWN - UP )
        move.w  d1,_deltay      * und fuer C-Programm ablegen
        move.w  d3,d1           * horizontale Werte nach d1
        lsr.b   #4,d3           * LEFT-Zaehlerwert in unteres Nibble
        and.w   #$000f,d1       * LEFT-Zaehlerwert ausblenden
        sub.w   d3,d1           * Delta-X Wert berechnen ( RIGHT. - LEFT )
        cmp.w   #$02,d1         * Delta-X Wert korrigieren ?
        bge     maxxkor         * ja -> dann verdoppeln
        cmp.w   #$fffe,d1
        bgt     nomaxx          * nein -> nicht korrigieren
maxxkor:
        lsl.w   #1,d1           * Delta-X Wert mal 2 nehmen
nomaxx:
        move.w  d1,_deltax      * und fuer C-Programm ablegen
*
* Entschluessle Tasten
*
        clr.w   d3              * d3 und
        move.w  d3,d4           * d4 loeschen
        btst    #8,d2           * Taste0 gedrueckt ?
        seq     d3              * ja -> d3 ungleich 0
        btst    #9,d2           * Taste1 gedrueckt ?
        seq     d4              * ja -> d4 ungleich 0
        lea     _tast0,a0
        move.w  d3,(a0)+        * Taste0 fuer C-Programm ablegen
        move.w  d4,(a0)         * Taste1 fuer C-Programm ablegen
        jsr     getreg          * hole alte Register ausser A7 wieder
        rts                     * zurueck ir C-Programm

        .page

****************************************
*
*     --------Subroutine's-------
*
****************************************
*
```

```
* setze oder loesche Bildschirmpunkt
*
* aus mc 12/85 seite 56
*
* uebergaberegister: d2 = berechnete x-koordinate (0 BIS 639)
*                    d3 = berechnete y-koordinate (0 BIS 199)
*                    clrflg = 2 -> Pixel invertieren
*                    clrflg = 1 -> Pixel loeschen
*                    clrflg = 0 -> Pixel setzen
*
setpnt:
        movea.l scrnbas,a0      * a0 mit basisadresse laden
        bsr     calct           * offset und bitnummer berechnen
        bhi     noset           * springe, wenn ausserhalb des bildschirmes
        cmp.w   #$1,clrflg      * Punkt loeschen ?
        bne     set             * nein ->
        bclr    d1,$0(a0,d0.w)  * ja
        bra     noset           * und return
set:
        cmp.w   #$2,clrflg      * Pixel invertieren ?
        bne     set1            * nein -> Punkt setzen
        bchg    d1,$0(a0,d0.w)  * ja
        bra     noset           * und return
set1:
        bset    d1,$0(a0,d0.w)  * setze Pixel auf Bildschirm

noset:  rts                     * zurueck in aufrufende Routine

*
* berechne adresse von bildschirmpunkt
*
calct:
        cmp.w   #maxx,d2        * x-koordinate zu gross ?
        bhi     calctq          * ja
        cmp.w   #maxy,d3        * y-koordinate zu gross ?
        bhi     calctq          * ja
        move.w  d3,d0           * in d0: 2048*(y mod 8) berechnen
        ror.w   #5,d0           * 2048 = 2^11 = 2^(16-5)
        and.w   #$3800,d0
        move.w  d3,d1           * in d1: 80 * int(y/8) berechnen
        lsr.w   #3,d1           * 8 = 2^3
        mulu    #80,d1
        add.w   d1,d0           * in d0 steht die summe
        move.w  d2,d1           * in d1: int(x/8) berechnen
        lsr.w   #3,d1

        add.w   d1,d0           * und zu d0 addieren
        move.w  d2,d1           * und in d1 bitnummer berechnen
        not.w   d1              * bitnummer = 7 - (x mod 8)
        cmp.b   d0,d0           * z-flag loeschen
calctq: rts                     * mit D0 = Offset und D1 = Bitnummer

        .page
*
* zeichnen einer linie
*
* Uebergaberegister:
*
*               d2,d3: startkoordinaten x1,y1
*               a2,a3: endkoordinaten   x2,y2
*
* verwendete Register:
*  d2,d3,d4,d6,d7,a2,a3,a4,a5,a6
*
draw:
        move.l  a2,d6           * in d6: DX = x1-x0 berechnen
        sub.l   d2,d6
        move.l  d6,a4           * XX = DX
        bge     posx            * springe, wenn XX >= 0
        neg.l   d6              * nein, XX := |DX|
*
* a4 = DX
* d6 = XX = ABS(DX)
* d2 = X1        d3 = Y1
*
posx:   move.l  a3,d7           * in d7: DY = y1-y0 berechnen
        sub.l   d3,d7
        move.l  d7,a5           * YY = DY
        bge     posy            * springe, wenn YY >= 0
        neg.l   d7              * nein, YY := |DY|
*
* a4 = DX               ; a5 = DY
* d6 = XX = ABS(DX)     ; d7 = YY = ABS(DY)
* d2 = X1               ; d3 = Y1
*
posy:
        cmpa.w  #0,a4           * Richtung von DX ?
        beq     testdy          * DX = 0 -> dann Waagerechte
        blt     negdx           * DX < 0 ->
        movea.w #1,a4           * DX > 0
        bra     testdy
negdx:  movea.w #-1,a4          * DX < 0
testdy:
        cmpa.w  #0,a5           * Richtung von DY ?
        beq     senkr           * DY = 0 -> dann Senkrechte
        blt     negdy           * DY < 0 ->
        movea.w #1,a5           * DY > 0
        bra     senkr

negdy:  movea.w #-1,a5          * DY < 0
*
* a4 = SGN(DX)          ; a5 = SGN(DY)
* d6 = XX = ABS(DX)     ; d7 = YY = ABS(DY)
* d2 = X1               ; d3 = Y1
*
senkr:
        clr.l   a6              * Flag = 0
        cmp.w   d6,d7           * |YY| < |XX| ?

        blt     grad45          * ja ->
        movea.w #1,a6           * Flag = 1
        exg     d6,d7           * d6 = XX = ABS(DY), d7 = YY = ABS(DX)
grad45:
        move.l  d6,d4           * Schleifenzaehler setzen
        move.w  d6,a2           * XX umspeichern
        move.w  d7,a3           * YY umspeichern
        lsr.w   #1,d6           * Differenz = (XX/2)
        move.l  a6,d7           * Flag nach d7
        bsr     setpnt          * setze ersten Pixelpunkt
```

```
*
* a2 = XX = ABS(DX)     ; a3 = YY = ABS(DY)
* a4 = SGN(DX)          ; a5 = SGN(DY)
* d2 = X1               ; d3 = Y1
* d4 = Schleifenzaehler (SP)
* d6 = Differenzsumme der Bewegungsrichtung (DSUM)
* d7 = Flag ob |Zeichenwinkel| > 45 (FL)
*
* Ab hier findet eigentliches Zeichnen statt
*
next:
        add.w   a3,d6           * DSUM = DSUM + YY
        cmpa.w  d6,a2           * XX > DSUM ( Linie fertig) ?
        bgt     xstep           * ja ->
        sub.w   a2,d6           * DSUM = DSUM - XX
        add.w   a5,d3           * Y1 = Y1 + SGN(DY)
        tst.l   d7              * Flag = 1 ?
        beq     xstep           * nein ->
        add.w   a4,d2           * X1 = X1 + SGN(DX)
        bra     point           * Setze Pixel an X1,Y1
xstep:
        tst.l   d7              * Flag = 1 ?
        bne     ystep           * ja ->
        add.w   a4,d2           * X1 = X1 + SGN(DX)
        bra     point           * Setze Pixel an X1,Y1

ystep:
        add.w   a5,d3           * Y1 = Y1 + SGN(DY)
point:
        bsr     setpnt          * setze oder loesche punkt an koordinate
        subq.w  #1,d4           * Schleifenzaehler erniedrigen
        bgt     next
        rts                     * routine verlassen

        .page
*
* rette alle Register ausser A7
*
savreg:
        movem.l d0-a6,regstack
        clr.l   d2
        clr.l   d3
        clr.l   a2
        clr.l   a3
        clr.w   clrflg          * loesche Flag fuer Zeichnen oder Loeschen
        rts

*
* hole alle Register wieder ausser A7
*
getreg:
        movem.l regstack,d0-a6

        rts

        .data
        .even
        .page
*
* Zwischenspeicher fuer alle Register ausser A7
*
regstack:
        ds.l    15              * Zwischenspeicher fuer Register
*
* Zwischenspeicher fuer Grafikroutinen
*
x:      ds.w    1               * Ablage von X-Koordinate fuer Grafikroutinen
y:      ds.w    1               * Ablage von Y-Koordinate fuer Grafikroutinen
clrflg: ds.w    1               * Flag fuer Zeichnen oder Loeschen
*
* Hier werden Uebergabewerte fuer C-Programm von Mouse abgelegt
*
_deltay: dc.w   0               * Bewegung in Y-Richtung
_deltax: dc.w   0               * Bewegung in X-Richtung
_tast0:  dc.w   0               * Wert von Taste0 (0 -> nicht gedrueckt)
_tast1:  dc.w   0               * Wert von Taste1 (0 -> nicht gedrueckt)

        .end
```

```
/*************************************************************/
/*                                                           */
/* Datei: MOUSE.C                                            */
/* --------------                                            */
/*                                                           */
/*************************************************************/

#include <stdio.h>
#include <ctype.h>

/* BIOS-Konstanten */
#define dircio      6    /* direct console I/O  */

/* Rechnerspezifische Steuerzeichen */
#define clrcrt 12       /* Bildschirm loeschen */
#define cr     13       /* <cr> */
#define lf     10       /* <lf> */
#define esc    27       /* <esc> */

/* Konstanten fuer Assemblerroutinen von Grafikroutinen */

#define pixset 0        /* Code, dass Pixel gesetzt wird      */
#define pixclr 1        /* Code, dass Pixel geloescht wird    */
#define pixchg 2        /* Code, dass Pixel invertiert wird   */

/* Externe Aufrufe:                        definiert in Modul: */
EXTERN LONG bdos();      /* Direkter BDOS Aufruf   ASM.O      */
EXTERN VOID exit();      /* Return zu CP/M         CLIB       */
EXTERN VOID drawlin();   /* Zeichne Linie          MOUSEC.S   */
EXTERN VOID setpoint();  /* Setze/Loesche Pixel    MOUSEC.S   */
EXTERN VOID kreuz();     /* Zeichne/Loesche Kreuz  MOUSEC.S   */
EXTERN VOID cardini();   /* Init Karte in Slot #2  MOUSEC.S   */
```

Bild 4. Dieses Beispielprogramm macht es möglich, Linien mit der Maus auf den Bildschirm zu zeichnen

```
/* Externe Variablen:   */
EXTERN int deltax;        /* eingelesener Mouse-Weg in X-Richtung */
EXTERN int deltay;        /* eingelesener Mouse-Weg in Y-Richtung */
EXTERN int tast0;         /* eingelesener Mousetasten-Wert 0      */
EXTERN int tast1;         /* eingelesener Mousetasten-Wert 1      */

int xmsalt;     /* Alte Mouse X-Position                */
int ymsalt;     /* Alte Mouse Y-Position                */
int xmsneu;     /* Neue Mouse X-Position                */
int ymsneu;     /* Neue Mouse Y-Position                */

/* liest ein Zeichen von Console */
UWORD input()
{
  return(bdos(dircio,(UWORD)0xff));
} /* input() */

/* Frage Console ab */
UWORD constat()
{
  return(bdos(dircio,(UWORD)0xfe));
} /* constat() */

/* Positionierung von Cursor auf Bildschirm */
/* gilt fuer MC68000-Rechner */
VOID goxy(xposcsr,yposcsr)
WORD xposcsr,yposcsr;
{
  putchar(esc);
  putchar('Y');
  putchar(xposcsr+32);
  putchar(yposcsr+32);
} /* goxy */

/* Lies Mousewerte ein und aktualisiere neue Pixelposition */
VOID rdcurpos()
{
 rdmouse();                    /* Lies Mousewerte ein         */
 xmsneu = xmsalt+deltax;       /* berechne neue X-Position    */
 ymsneu = ymsalt+deltay;       /* berechne neue Y-Position    */
 /* korrigiere X-/Y-Position, wenn ausserhalb von Bildschirm  */
 if (xmsneu < 0)
   xmsneu = 0;
 if (xmsneu > 639)
   xmsneu = 639;
 if (ymsneu < 0)
   ymsneu = 0;
 if (ymsneu >199)
   ymsneu = 199;
} /* rdcurpos */

/* Bewege Kreuz auf Bildschirm, bis irgendeine Mousetaste     */
/* oder Tastatur gedrueckt                                    */
/* Bei Aufruf existiert kein Kreuz auf Bildschirm.            */
/* Ebenso nach Verlassen der Routine                          */
UBYTE befehlin()
{
 UBYTE keypress;        /* Status von Tastatur  */
  {
   kreuz(xmsalt,ymsalt,pixchg);        /* setze Kreuz an mom. Position */

   befloop:    /* Schleife fuer Mouse- und Tastaturabfrage */

    rdcurpos();     /* Lies Mousewerte ein und aktualisiere X-/Y-Position */
    keypress = constat(); /* Frage Tastaturstatus ab */
    if ((tast0 != 0) || (tast1 != 0) || (keypress !=0))
    /* Wenn irgendeine Tast gedrueckt wurde */
     {
      kreuz(xmsalt,ymsalt,pixchg);      /* Loesche altes Kreuz */
      xmsalt = xmsneu;  /* Cursorposition aktualisieren */
      ymsalt = ymsneu;  /*     dito                     */
      if (tast0 != 0) return(255);
      else
      if (tast1 != 0) return(254);
      else
       return(input());
     }
    else
     {
      if ((xmsneu != xmsalt) || (ymsneu != ymsalt))
      /* Wenn sich Position veraendert hat */
       {
        kreuz(xmsalt,ymsalt,pixchg);      /* Loesche altes Kreuz */
        kreuz(xmsneu,ymsneu,pixchg);      /* Zeichne neues Kreuz */
        xmsalt = xmsneu;  /* Cursorposition aktualisieren */
        ymsalt = ymsneu;  /*     dito.                    */
       }
     }
   goto befloop;
  }

} /* befehlin() */

/* Bewege solange Linie von uebergebenen Anfangskoordinate, bis        */
/* Taste0 wieder losgelassen wird */
VOID movline(xanfang,yanfang,clrflg)
register int xanfang,yanfang;   /* Anfangskoordinaten */
register int clrflg;   /* Flag fuer Zeichnen oder Loeschen von Linie */

{
 register int x2,y2;   /* Laufvariable der Endkoordinaten      */
 kreuz(xanfang,yanfang,clrflg); /* setze Kreuz an Anfangspunkt */
 x2 = xanfang; /* Am Anfang: Endposition = Anfangsposition     */
 y2 = yanfang; /*               dito                           */
 setpoint(xanfang,yanfang,clrflg); /* Anfangspunkt setzen      */
 do /* schiebe solange Linie auf Bildschirm bis Taste0 losgelassen */
  {
   rdcurpos(); /* Lies Mouseposition nach xmsneu und ymsneu ein */
   /* Wenn sich Mouseposition veraendert hat */
   if ((xmsneu != xmsalt) || (ymsneu != ymsalt))
    {
```

```
      drawlin(xanfang,yanfang,x2,y2,clrflg); /* Loesche alte Linie */
      kreuz(x2,y2,clrflg); /* Loesche altes Kreuz an alter Endkoordinaten */
      x2 = xmsneu;                          /* neue X-Endkoordinate */
      y2 = ymsneu;                          /* neue Y-Endkoordinate */
      drawlin(xanfang,yanfang,x2,y2,clrflg); /* Zeichne neue Linie */
      kreuz(x2,y2,clrflg); /* Setze Kreuz an Endkoordinaten */
      xmsalt = xmsneu;  /* Aktualisiere X-/Y-Position fuer rdcurpos() */
      ymsalt = ymsneu;
     }
   } while (tast0 != 0); /* Solange Taste0 gedrueckt */
  kreuz(x2,y2,clrflg); /* Loesche Kreuz an Endkoordinatenpunkt  */
  if ((xanfang == x2) && (yanfang == y2))
   setpoint(xanfang,yanfang,clrflg); /* Wenn keine Linie gezeichnet   */
} /* movline */


/*==============================*/
/*                              */
/*   Beginn von Hauptprogramm   */
/*                              */
/*==============================*/

main()
{
 int entpr;       /* Tastenentprellung                                  */
 int wait;        /* Wartezeit bei Ausgabe einer Meldung auf Bildschirm */
 int keychr;      /* Tastatur-Wert                                      */
 UWORD command;   /* Eingabevariable                                    */

 /* Initialisiere Variablen */
 keychr = 0;        /* Tastaturwert zuruecksetzen */

 putchar(clrcrt);       /* loesche Bildschirm */
 goxy(10,10);
 printf("Soll Mousekarte in Slot #2 initialisiert werden: ");
 do
  {
   keychr=input();      /* Lies Zeichen ein */
   keychr=toupper(keychr);
  } while ((keychr != 'J') && (keychr != 'N'));

 if (keychr == 'J')

  {
   printf("%c\n\r",keychr); /* Echo Eingabe auf Bildschirm */
   printf("\n\r Karte in Slot #2 wird initialisiert");
   cardini();           /* Initialisiere Karte in Slot #2         */
   for (wait = 0; wait < 20000; wait++)
   ; /* Warteschleife fuer obige Textausgabe */
  }
 keychr = 0;             /* Tastaturwert zuruecksetzen */
 printf("%cCG\n",esc);  /* umschalten in Grafikmodus   */
 putchar(clrcrt);        /* loesche Bildschirm         */
 xmsalt = 319;           /* Bildschirmmitte X-Position        */
 ymsalt = 100;           /* Bildschirmmitte Y-Position        */
 do /* Zeichne solange Linien, bis "Q" oder "q" gedrueckt */
  {
   /* Bewege Kreuz, bis irgendeine Mousetaste oder Tastatur getaetigt */
   command = befehlin();
   if (command == 255) /* Wenn Linie gezeichnet werden soll */
    {
     for (entpr = 0; entpr < 1000; entpr++)
     ; /* Tastenentprellung */
     /* Schiebe solange Linie auf Bildschirm, bis Taste0 losgelassen */
     /* Dabei wird immer Kreuz an aktueller Mouseposition gesetzt !! */
     movline(xmsneu,ymsneu,pixchg);
     for (entpr = 0; entpr < 1000; entpr++)
     ; /* Tastenentprellung */
    } /* Taste0 ist gedrueckt gewesen */
  } while ((command != 'Q') && (command != 'q'));
 printf("%cCT\n",esc);                  /* umschalten in Textmodus */
 exit();
} /* main */
```

liefern die Anzahl der Impulse für jede Richtung. Um den Relativbetrag der Bewegung in horizontaler und vertikaler Richtung festzustellen, müssen die relativen Zählerstände einer Ebene (Rechts-Links und Unten-Oben) voneinander abgezogen werden. Der eingelesene Mausweg in X-Richtung wird dann in _deltax, der eingelesene Mausweg in Y-Richtung in _deltay abgelegt. Der Zustand der rechten ATARI-Maustaste wird in _tast0, der der linken Maustaste in _tast1 abgelegt. Gleichzeitig werden die Zähler für die nächste Abfrage wieder zurückgesetzt.

Bild 4 zeigt eine kleine Beispielroutine, die ein Kreuz solange auf dem Bildschirm bewegt, bis von der Tastatur „Q“ eingegeben wurde. Wird die rechte Maustaste gedrückt und gehalten, wird solange eine „Gummilinie“ gezeichnet, bis sie wieder losgelassen wird. Danach bleibt diese Linie auf dem Bildschirm und man kann das Kreuz an eine andere Stelle bewegen, um dort mit dem Zeichnen einer neuen Linie zu beginnen.

Erstellen eines lauffähigen Programmes

Um ein lauffähiges Testprogramm zu erhalten, gibt man folgende Befehle nacheinander ein:

```
AS68 MausC.S
C Maus
CLINK Maus MausC ASM
```

Wurde alles fehlerfrei assembliert, muß eine lauffähige Datei Maus.68k erstellt worden sein. Nach dem Start dieser Datei erscheint die Frage, ob die Karte in Steckplatz x initialisiert werden soll. Bei Eingabe von J oder j wird die Karte in der Scratch Area eingetragen. Bei Eingabe von N oder n wird davon ausgegangen, daß die Karte schon in der Scratch Area existiert. Sie wird deshalb nicht nochmal eingetragen. Danach beginnt dann das eigentliche Testprogramm.

Durch die einfache Rechnerschnittstelle dürften keine Probleme bei der Anpassung an andere 68000-Systeme auftreten. Bei der Anpassung an andere Prozessortypen muß die Adreßdecodierung und Select-Logik angepaßt werden. Die Zähler-Schaltung kann ohne Änderung übernommen werden. Die Einleseroutine ist natürlich rechnerspezifisch und muß für andere Prozessortypen entsprechend umgeschrieben werden. Das gleiche gilt für die Initialisierung der Karte.

Literatur

[1] *Harald Astheimer:* Grafik in C, mc Heft 12/ 1985 Seite 93.

Thomas Schlenger-Klink

Programmierbare Logikbausteine

Teil 2: Basic-EMUF programmiert GALs

Mit einer kleinen Zusatz-Platine können die beiden GAL-Bausteine 16V8 und 20V8 am Basic-EMUF programmiert werden. Die Ansteuerung durch einen IBM-PC ist ebenfalls möglich. Das Pascal-Programm zu diesem Beitrag benutzt das Centronics-Port. Es ist auf Diskette zu haben.

In der vorigen Ausgabe haben wir das Innere der GAL-Bausteine 16V8 und 20V8 geschildert. Zur Programmierung dieser EEPLDs (Electrically Erasable Programmable Logic Devices) reicht eine relativ einfache Schaltung. Alle dann notwendigen Signale können mit einem geeigneten Programm erzeugt werden. Die Ansteuerung des mc-GAL-Programmers kann ein mc-Basic-EMUF übernehmen. Es reicht aber auch der Centronics-Anschluß eines IBM-PCs (oder kompatiblen Rechners) aus.

Die Schaltung

Bild 1 zeigt die Schaltung des Programmierzusatzes. Der Baustein TL497 (IC1) erzeugt die Programmierspannung von 16.5 Volt. Mit dem Potentiometer P1 kann man diese Spannung genau einstellen. Mit der Schaltstufe aus TR2 und TR3 wird diese Programmierspannung bei Bedarf auf das zu programmierende GAL gelegt. Um die GALs beim Einsetzen und Herausnehmen vor Beschädigungen zu schützen, kann durch eine weitere Schaltstufe (TR1, TR2) die Betriebsspannung der Programmiersockel abgeschaltet werden. Die Belegungen der Steuersignale am PIO-Baustein auf dem Basic-EMUFs ist in *Tabelle 1* gezeigt. Für den Centronics-Anschluß zeigt *Tabelle 2* die Signale. Wenn man mit dem Basic-EMUF arbeiten will, muß man dort den Uhrenbaustein entfernen, denn da kann es zu Störungen kommen.

Das Layout der Platine ist in den *Bildern 2 und 3* gezeigt, der Bestückungsplan in *Bild 4.* In *Tabelle 4* ist die Stückliste des GAL-Programmers angegeben. Die Platine kann auf verschiedene Arten an den Basic-EMUF angeschlossen werden:

1. Lötet man von der Lötseite her an ST2 eine 40polige zweireihige Buchsenleiste an, so kann die Platine als „Huckepack"-Modul auf den EMUF aufgesteckt werden.

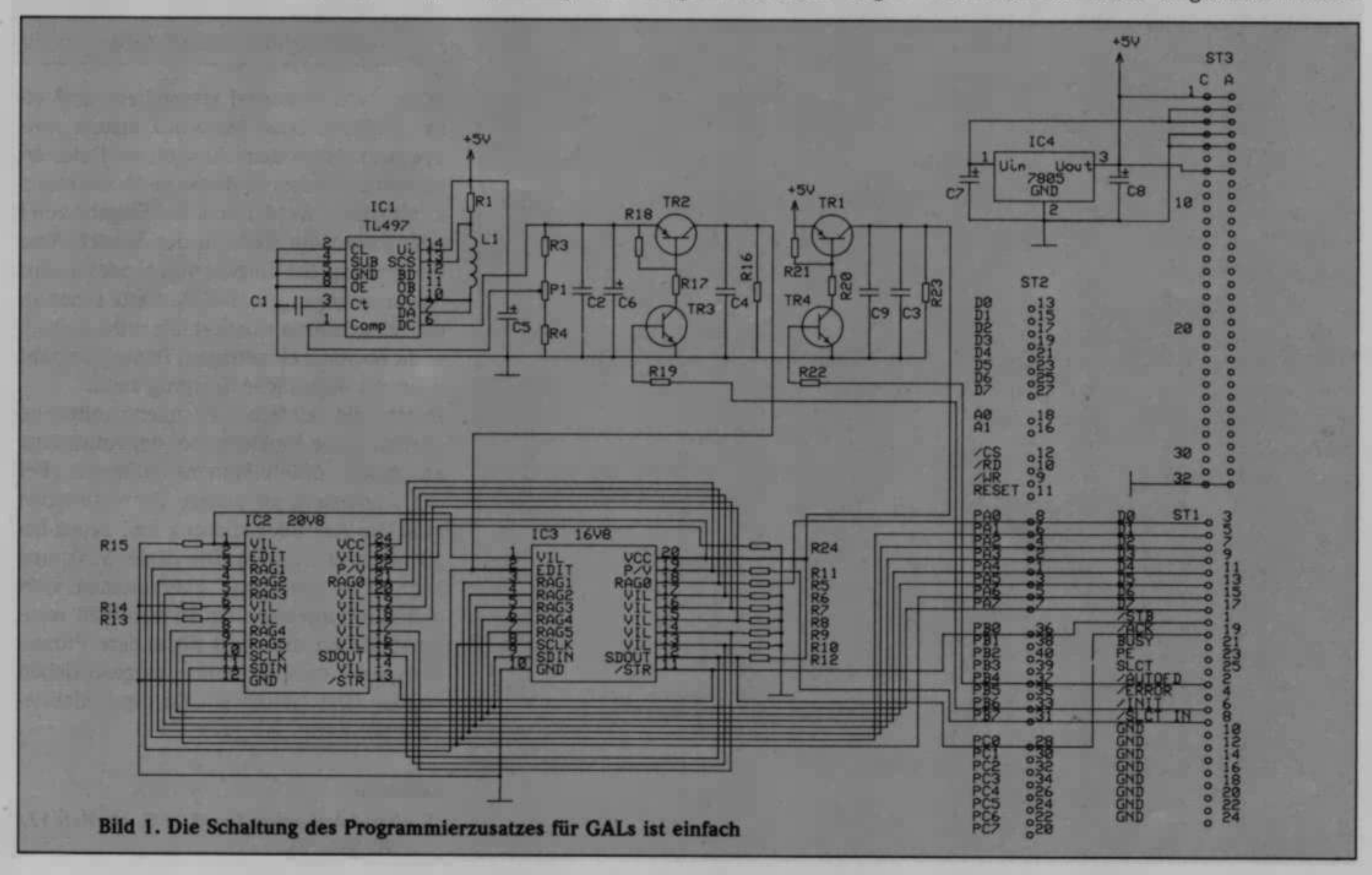


Bild 1. Die Schaltung des Programmierzusatzes für GALs ist einfach

Das ist allerdings bei einem vollbestückten Basic-EMUF nicht möglich, weil dafür einige Bauelemente auf dem EMUF zu hoch sind.

2. Normalerweise wird die Verbindung über ein 40poliges Flachbandkabel hergestellt werden. Dabei sollte dieses Kabel nicht länger als 25 cm sein, damit die Signale sauber bleiben.

Beim Aufbau löten Sie zunächst alle Widerstände und die Spule L1 ein. Danach die Kondensatoren, IC-Sockel, Transistoren und alle anderen Bauteile. Auf der Karte gibt es um die Programmiersockel ausreichend Platz. Dieser Platz ist für Nullkraft-Sockel vorgehalten, die mehr Raum einnehmen als normale IC-Sockel. Nullkraft-Sokkel kann man leider nicht überall kaufen. Außerdem sind sie relativ teuer. Für sparsame Leute empfehlen wir daher die Verwendung eines IC-Sockels mit Präzisionskontakten, in den ein einziger 24pol. Nullkraftsockel eingelötet wird. Dieses Sockelgebilde kann in jeden der beiden unterschiedlich großen Sockel auf der Platine eingesteckt werden. Wenn Sie aber nur selten GAL-Bausteine programmieren wollen, so reichen auch ganz normale Präzisionssockel aus.
Der Spannungsregler (IC 4) kann die Stromversorgung des Basic-EMUFs mit übernehmen. Dann muß er aber mit einem entsprechenden Kühlkörper versehen werden. Bei Stromversorgung aus einem bereits stabiliserten Netzgerät, darf der Spannungsregler nicht eingesetzt werden. Nach dem Einlöten der Bauelemente sollte die Platine nochmals sehr sorgfältig auf Löt- und Bestückungsfehler hin überprüft werden.

Der Abgleich des Spannungswandlers

Zum Abgleich des Spannungswandlers IC 1 darf der Basic-EMUF nicht angeschlossen sein. Unter Betriebsspannung sollte dann zunächst ein Strom von ca. 10 mA fließen (IC 1 bestückt). Zum Abgleich wird zwischen die Anschlüsse 2 (EDIT) und 10 (GND) beim 20-poligen Sockel (IC 3 = 16V8) ein Widerstand von etwa 3,3 kOhm gesteckt. An den Anschluß 37 des PIO-Steckers ST2 wird eine Brücke zur +5 Volt-Spannung gelegt. Nach dem Einschalten der Betriebsspannung kann die Programierspannung am Anschluß 2 von IC 2 oder IC 3 auf exakt 16,5 Volt eingestellt werden. Wenn die Brücke und der Belastungswiderstand wieder entfernt sind, ist der Programmierer fertig.

Tabelle 1: ST2 entspricht ST4 des Basic-EMUFs

8	PA 0	RAG 0
6	PA 1	RAG 1
4	PA 2	RAG 2
2	PA 3	RAG 3
1	PA 4	RAG 4
3	PA 5	RAG 5
5	PA 6	SCLK
7	PA 7	SDIN
36	PB 0	/STR
37	PB 4	VCC ein, wenn "HIGH"
33	PB 6	VPP ein, wenn "HIGH"
31	PB 7	Prog/Verify Programm, wenn "HIGH"
28	PC 0	SDOUT

Tabelle 2: ST1 ist der Centronics-Anschluß zum IBM-PC

3	D0	RAG 0
5	D1	RAG 1
7	D2	RAG 2
9	D3	RAG 3
11	D4	RAG 4
13	D5	RAG 5
15	D6	SCLK
17	D7	SDIN
1	SSTROBE	/STR
2	AUTOFD	VCC ein wenn "HIGH"
6	/INIT	VPP ein wenn "HIGH"
8	/SLCTIN	Prog/Verify Programm wenn "HIGH"
19	/ACK	SDOUT

Tabelle 3: Die Belegung der VG-Steckerleiste ST3

1	a,c	+5 Volt
2	a,c	Betriebsspannung ca. +9 Volt
3	a,c	Betriebsspannung ca. +9 Volt
4	a,c	GND
5	a,c	GND
7	a,c	+5 Volt
32	a,c	GND

Tabelle 4: Die Stückliste zum mc-GAL-Programmer

IC 1	TL497
IC 2	Programmiersockel 20pol.
IC 3	Programmiersockel 24pol.
IC 4	7805 Spannungsregler
R 1	10 Ω Kohleschicht, 3r
R 3	110 kΩ Metallschicht, 3r
R 4	8,2 kΩ Metallschicht, 3r
R 5 - R 24	10 kΩ Kohleschicht, 3r
P 1	1 kΩ Kohleschicht-Potentiometer
TR 1, TR2	BC 556B o.ä.
TR 3, TR4	BC 546B o.ä.
L 1	100 µH Drosselspule, z.B. DALE IM2
C 1	100 pF Kerko, 1r
C 2	100 nF Vielschicht, 1r
C 3	100 nF Vielschicht, 1r
C 4	100 nF Vielschicht, 1r
C 5	100 µF AL-Elko 25V, 2r
C 6	100 µF AL-Elko 25V, 2r
C 7	10 µF Tantal 16V, 1r
C 8	10 µF Tantal 16V, 1r
ST 1	26pol. Pfostenstecker, 2reihig
ST 2	40pol. Pfostenstecker, 2reihig
ST 3	VG-Messerleiste 64ac
Platine	GAL-PROG

Die Software

Zunächst muß der mc-GAL-Programmer jetzt getestet werden. Das Programmier-Programm für den Basic-EMUF (Listing gibt es auf Anfrage von der mc-Redaktion, eine Diskette für 25 DM ist ebenfalls erhältlich) ist so konzipiert, daß es in einem EPROM für den Assemblersokkel (IC8 auf Basic-EMUF) arbeitet. Hier zeigt sich wieder der Sinn des Namens EMUF: Einplatinen-Computer mit universeller Festprogrammierung. Da es bereits ein ganze Reihe von Programmierwerkzeugen zur Entwicklung von PALs gibt, wurde auf ein entsprechendes PAL-Assembler-Programm verzichtet. Vielleicht entwirft der eine oder andere Leser einen GAL-Assembler für den Basic-EMUF, was dann wohl das preiswerteste Entwicklungsgerät zur GAL-Programmierung ergäbe.
Die meisten Programmiersysteme (PAL-Assembler, CUPL, ABEL, LOGIC usw.) liefern eine Datei im JEDEC-Format. Aus dem Inhalt dieser Datei müssen die entsprechenden Bit-Muster zur Programmierung erzeugt werden. Die zusätzlichen Bits für das Architecture Control Word der GALs werden teilweise bereits von den Programmiersystemen geliefert oder können einfach hinzugefügt werden. In *Tabelle 5* sind die Personality-Werte für einige Standard-PALs aufgeführt. Durch Programmierung der

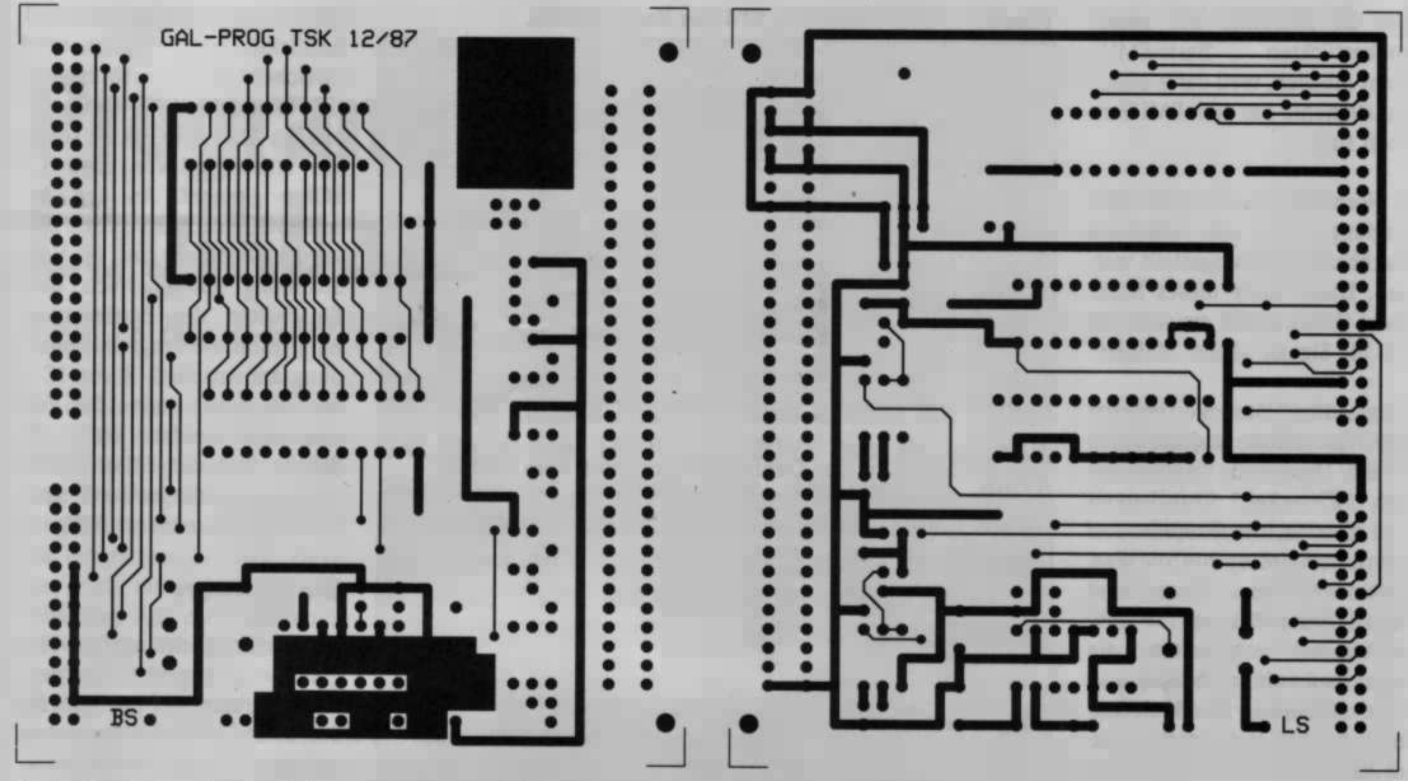


Bild 2. Das Layout der Bestückungsseite des mc-GAL-Programmers **Bild 3. Die Lötseite**

entsprechenden Ausgangspolarität können damit die jeweiligen „H"- bzw. „L"-PAL-Typen ersetzt werden. Das Basic-EMUF-GAL-Programmier-Programm wurde als Maschinenprogramm ausgearbeitet. Es nutzt die im 8052-Basic-AH-Baustein eingebaute Möglichkeit, eigene Befehle an den Basic-Befehlssatz anzuhängen. Dabei wurden folgende Befehle implementiert:

1. GALTYPE N. N gibt dabei den Typ des GALs an: 0=16V8, 1=20V8
2. GALREAD. Lesen eines GALs, Typ durch GALTYPE angegeben
3. GALPROG. Programmieren eines GALs, Typ durch GALTYPE angegeben
4. GALCOMP. Vergleichen eines GALs (Typ durch GALTYPE angegeben) mit dem Speicherinhalt.
Status: 0FFFFH = Vergleich fehlerfrei
<>0FFFFH = Bit-Adresse = Fehler beim Programmieren an der angegebenen Bit-Adresse
5. BULKERASE. Mit diesem Befehl wird ein GAL komplett gelöscht. Es wird keine Leerprüfung des GALs vorgenommen. Bei bereits benutzten GALs muß zuerst ein BULKERASE durchgeführt werden.
6. JLOAD. Lesen einer JEDEC-Dateie über die Terminal-Schnittstelle
7. BITREAD N. Lesen eines Bits der GAL-Matrix. (N=FUSE)
Status: 0 = Bit ist „LOW„
0FFFFH = Bit ist „HIGH"
8. BITSET N. Setzen eines Bits der GAL-Matrix. (N=FUSE)
9. RESBIT N. Rücksetzen eines Bits der GAL-Matrix. (N=FUSE) 10)
10) SECURITY. Setzen der Security-Fuse im GAL

Danach kann die GAL-Matrix nicht mehr ausgelesen werden.

Der JLOAD Befehl wertet nur Zeilen aus, die mit „*L" beginnen. Direkt dahinter muß die Adresse einer Sicherung aus der Matrix dezimal angegeben werden (z. B.

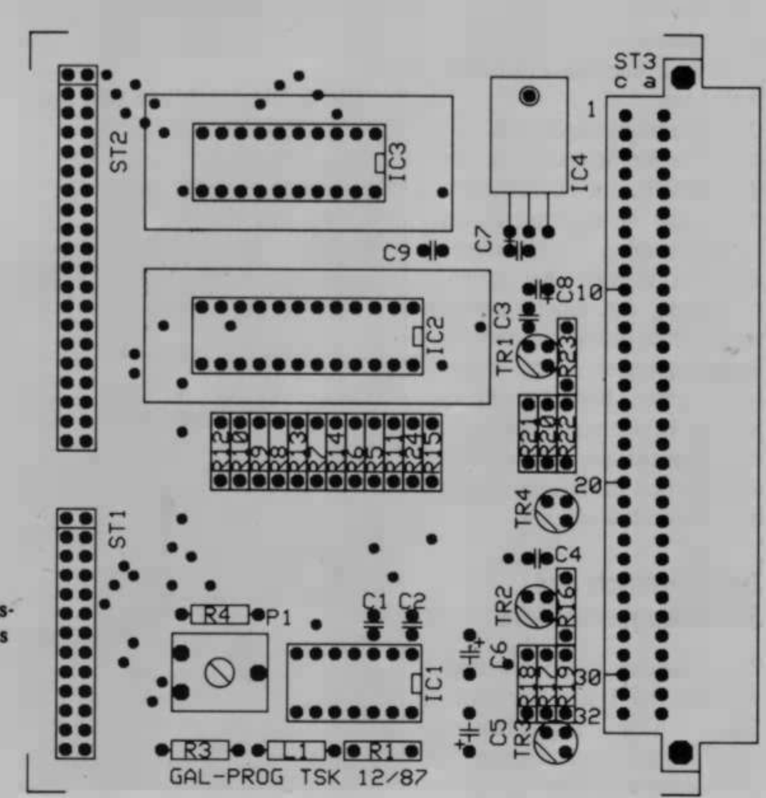


Bild 4. Der Bestückungsplan des Gerätes

Tabelle 5: Architecture Control Word für einige Standard-PALs

```
10P8
*L2048 00000000
*L2120 00000000
*L2128
1111111111111111111111111111111111111111111111111111111111111111
*L2192 10

14P4
*L2048 00000000
*L2120 11000011
*L2128
1111111111111111111111111111111111111111111111111111111111111111
*L2192 10

16P2
*L2048 00000000
*L2120 11100111
*L2128
1111111111111111111111111111111111111111111111111111111111111111
*L2192 10

16R4
*L2048 00000000
*L2120 11000011
*L2128
1111111111111111111111111111111111111111111111111111111111111111
*L2192 01

16R8
*L2048 00000000
*L2120 00000000
*L2128
1111111111111111111111111111111111111111111111111111111111111111
*L2192 01
```

Tabelle 7: Fuse-Adressen und ihre Bedeutung für GALs 16V8 und 20V8

```
                 Für GAL 16V8
0000 - 2047      Logik-Matrix
2048 - 2055      XOR-Bit für Ausgang 19 - 12
2056 - 2119      Elektronische Signatur UES = 64 Bit für eigene
                 Anwendungen
2120 - 2127      AC1-Bit für Ausgang 19 - 12
2128 - 2191      Produkt-Term Freigabe PT0 - PT63
2192             SYN-Bit
2193             AC0-Bit

                 Für GAL 20V8
0000 - 2559      Logik-Matrix
2560 - 2567      XOR-Bit für Ausgang 22 - 15
2568 - 2631      Elektronische Signatur UES = 64 Bit für eigene
Anwendungen
2632 - 2639      AC1-Bit für Ausgang 22 - 15
2640 - 2647      Produkt-Term Freigabe PT0 - PT63
2704             SYN-Bit
2705             AC0-Bit

Fuse = 0         Verbindung in der Matrix vorhanden
Fuse = 1         Verbindung in der Matrix nicht vorhanden
```

*L0256). Nach einem Leerzeichen folgen die Daten für diese und die folgenden Sicherungs-Nummern. Vor Beginn des Ladens wird der gesamte Zwischenspeicher für Sicherungen gelöscht (d.h. alle Sicherungen=0). Es müssen daher nur die Sicherungen, die auf 1 gesetzt sind, übertragen werden. Dies kann bei längeren Dateien die Ladezeit erheblich verkürzen. Eine Zeile mit „*C" am Anfang bricht den Ladevorgang ab. Alle weiteren Kommandos im JEDEC Format werden dann durch das Programm nicht mehr ausgewertet und ignoriert. In der vorliegenden Version wird keine Prüfsumme gebildet bzw. ausgewertet. Damit können vorhande JEDEC-Dateien, die zur Programmierung von PALs generiert wurden, durch Bearbeiten mit einem Texteditor um die zusätzlichen Informationen für die GAL Programmierung erweitert werden, ohne daß die Prüfsumme erneut berechnet werden muß. In *Tabelle 6* ist als Beispiel die JEDEC-Datei eines GALs für den Basic-EMUF gezeigt. Es ist die 32-KByte-Version des Adreßdekoders.

Die zusätzlichen Basic-Befehle, die das Programmieren bewirken, übergeben und erwarten ihre Daten auf dem Stack des 8052-AH-Basic-Systems. Die Gal-Matrix wird oberhalb von MTOP gespeichert. Der Wert dieser Variablen muß um mindestens 1024 Byte herabgesetzt werden, damit ausreichend Platz für die GAL-Daten vorhanden ist. Im *Bild 5* sind einige Beispielprogramme angegeben, in denen die Verwendung der zusätzlichen Befehle gezeigt ist.

Ein Programmier-Beispiel

Unter den vielen verschiedenen Möglichkeiten der Ausgangszellen-Programmierung soll hier als Beispiel der Ersatz eines 16P8 besprochen werden. Dieser PAL-Baustein besitzt maximal 8 Ausgänge, die jedoch mit Hilfe eines Produkttermes in den Tree-State-Zustand versetzt werden können, und dadurch zum Eingang werden. Die Adres-

Tabelle 6: JEDEC-Datei für den Adreßdecoder des Basic-EMUF

```
Name                 GAL32k
*QP20
*QF2194
*G0
*F0
*L0000 01111111111111111111111111111111
*L0256 01010111011101110111011101111010
*L0768 01100111011101110111011101111010
*L1024 01111111111111111111111111111111
*L1280 01111011111111111111111111111111
*L1536 01010111011101110111011101111001
*L1792 01100111011101110111011101111001
*L2048 10001000000000000000000000000000
*L2112 00000000001000000111111111111111
*L2144 11111111111111111111111111111111
*L2176 111111111111111110
*C205F
*A1F8
```

```
PAL10L8
VERSION 1.1 8KB RAM
8052 BASIC-EMUF
```

```
A7 A15 A14 A13 A12 A11 A10 A9 A8 GND
A6 -/RAMST /ADDA /CSB /CSR /PIO NC /PIOST /CSA VCC

PIO=A15*A14*A13*/A12*/A11*/A10*/A9*/A8*/A7*/A6
RAMST=A15*A14*A13*/A12*/A11*/A10*/A9*/A8*/A7*A6
PIOST=A15*A14*A13*/A12*/A11*/A10*/A9*/A8*A7*/A6
ADDA=A15*A14*A13*/A12*/A11*/A10*/A9*/A8*A7*A6
CSR=/A15*/A14*/A13
CSB=A15*/A14
CSA=/A15
DESCRIPTION
PIO     E000-E03F
RAMST   E040-E07F
PIOST   E080-E0BF
ADDA    E0C0-E0FF
CSR     0000-1FFF
CSB     8000-BFFF
CSA     0000-7FFF
```

sen der Sicherungen beziehen sich auf ein GAL vom Typ 16V8. In *Tabelle 7* sind die Bedeutungen nochmals zusammengefaßt.

Das erste Bit ab Adresse 2048 ist für die Polarität von Ausgang 19 verantwortlich, das nächste für Ausgang 18 usw. Ist das entsprechende Bit 0, so wird der Ausgang aktiv LOW. Sollen zum Beispiel Ausgang 19,17 akt. Low und Ausgang 18,16,15,14,13,12 akt. High gesetzt werden, dann wird das mit folgender Zeile erreicht:

„*L2048 01011111"

Das SYN-Bit auf Adresse 2192 gibt im LOW-Zustand die D-Flipflops der Ausgangszelle frei. Da bei einem 16P8 keine Register vorhanden sind, muß dieses Bit auf 1 gesetzt werden. Das AC0-Bit jeder Ausgangszelle auf Adresse 2193 bestimmt zusammen mit dem entsprechenden AC1-Bit die Freigabe jedes Ausgangspuffers.

Setzt man AC0 und AC1
0 0, so wird der Ausgang dauernd freigegeben, kein Tree-State möglich.
0 1, so wird der Ausgang dauernd Tree-State, er wird zum Eingang.
1 0, so bewirkt das die Freigabe des Ausgangspuffers durch PIN 11
1 1, so bewirkt das die Freigabe des Ausgangspuffers Produktterm 1 der entsprechenden Ausgangszelle

Das erste Bit ab Adresse 2120 ist das AC1-Bit von Ausgang 19, das nächste für Ausgang 18 usw. Für den Typ 16P8 müssen AC0 und alle AC1-Bits auf 1 gesetzt sein, d.h. „*L2120 11111111" und „*L2192 11".

Die 64 Bit ab Adresse 2128 geben im „1"-Zustand den entsprechenden Produktterm (PT) frei, Bit 2128 für PT0, Bit 2129 für PT1 usw. Für den 16P8 können alle Terme freigegeben werden, also

*L2128 1111111111111111111111111 1111111111111111111111111 1111 111111

Die Bits der Sicherungen müssen entsprechend der Logikfunktion programmiert werden. Die Zuordnung ist in Teil 1 dieses Artikels [4] geschildert. Soll eine Verbindung zwischen den Matrixkreuzungen vorhanden sein, so ist das entsprechende Bit auf „0" zu setzen.

Nach Festlegung der Ausgänge und der logischen Gleichungen, wird die Programmierung auf folgende Weise durchgeführt:

```
10      REM BEISPIEL GAL-PROGRAMMER
20     MTOP=7000 :  REM SPEICHER FREIMACHEN
30      CALL 2048H :  REM WENN LCDKEY ANGESCHLOSSEN
40      GALTYPE 0 :  REM TYP 16V8
100     PRINT "QUELLE: (J)EDEC-FILE ODER (G)AL (J/G) ? ",
110     DO  : CH=GET :  UNTIL CH<>0
120     PRINT  :  IF CH<>74 THEN 150
130     JLOAD
140     GOTO 500
150     IF CH<>71 THEN 100
160     PRINT "GAL EINSETZEN (QUELLE)   ",
170     GOSUB 1000
180     GALREAD  :  REM GAL EINLESEN
500     REM GAL-PROGRAMMIEREN
510     PRINT "GAL EINSETZEN (ZIEL)     ",
520     GOSUB 1000
530     BULKERASE  :  REM GAL LOESCHEN
540     GALPROG  :  REM GAL PROGRAMMIEREN
550     GALCOMP  :  REM VERGLEICHEN
560     POP V :  IF V<>0FFFFH THEN  PRINT "FEHLER BEI DER
PROGRAMMIERUNG"
999     END
1000    PRINT "WEITER MIT <RETURN>",
1010    DO  : CH=GET :  UNTIL CH=13
1020    PRINT
1030    RETURN

>RUN

QUELLE: (J)EDEC-FILE ODER (G)AL (J/G) ? J
GAL EINSETZEN (ZIEL)     WEITER MIT <RETURN>

READY
>RUN

QUELLE: (J)EDEC-FILE ODER (G)AL (J/G) ? G
GAL EINSETZEN (QUELLE)   WEITER MIT <RETURN>
GAL EINSETZEN (ZIEL)     WEITER MIT <RETURN>

READY

.PA
10      REM BEISPIEL GAL INHALT ZEIGEN
20     MTOP=7000 :  REM SPEICHER FREIMACHEN
30      CALL 2048H :  REM WENN LCDKEY ANGESCHLOSSEN
40      GALTYPE 0 :  REM TYP 16V8
100     GALREAD  :  REM GAL EINLESEN
200     REM FUSEMATRIX ZEIGEN
210     IF (XBY(MTOP+10)=1) THEN L=40 ELSE L=32 :  REM GALTYP
220     FOR I=0 TO 63
230     PRINT USING(##),I, :  PRINT USING(0),"   ",
240     FOR J=0 TO L-1
250     BITREAD 32*I+J :  POP X :  IF X THEN  PRINT "-", ELSE  PRINT
"X",
260     NEXT J
270     PRINT
280     NEXT I
290     PRINT
400     PRINT "XOR      ", : N1=0 : N2=7 :  GOSUB 2000
410     PRINT "UES      ", : N1=8 : N2=63 :  GOSUB 2000
420     PRINT "AC1      ", : N1=72 : N2=7 :  GOSUB 2000
430     PRINT "PTD      ", : N1=80 : N2=63 :  GOSUB 2000
440     PRINT "SYN AC1 ", : N1=144 : N2=1 :  GOSUB 2000
999     END
2000    REM BITS EINZELN ANZEIGEN
2010    FOR I=(L*64+N1) TO (L*64+N1+N2)
2020    BITREAD I :  POP X :  IF X THEN  PRINT "1", ELSE  PRINT "0",
2030    NEXT I
2040    PRINT
2050    RETURN
.PA
RUN

  0     X-------------------------------
  1     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  2     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  3     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  4     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  5     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  6     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  7     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  8     X-X-X---X----X---X---X---X---X-X
  9     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 10     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Bild 5. Ein Beispielprogramm in Basic-EMUF-Basic zur Bedienung des mc-GAL-Prommers

```
 11    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 12    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 13    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 14    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 15    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 16    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 17    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 18    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 19    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 20    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 21    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 22    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 23    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 24    X--XX---X----X---X---X---X---X-X
 25    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 26    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 27    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 28    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 29    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 30    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 31    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 32    -X---X---X----------------------
 33    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 34    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 35    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 36    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 37    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 38    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 39    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 40    X----X--------------------------
 41    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 42    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 43    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 44    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 45    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 46    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 47    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 48    X-X-X---X----X---X---X---X---XX-
 49    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 50    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 51    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 52    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 53    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 54    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 55    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 56    X--XX---X----X---X---X---X---XX-
 57    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 58    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 59    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 60    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 61    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 62    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 63    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XOR     10000000
UES
0000000000000000000000000000000000000000000000000000000000000000
AC1     00100000
PTD
1111111111111111111111111111111111111111111111111111111111111111
SYN AC1 10

READY
>
```

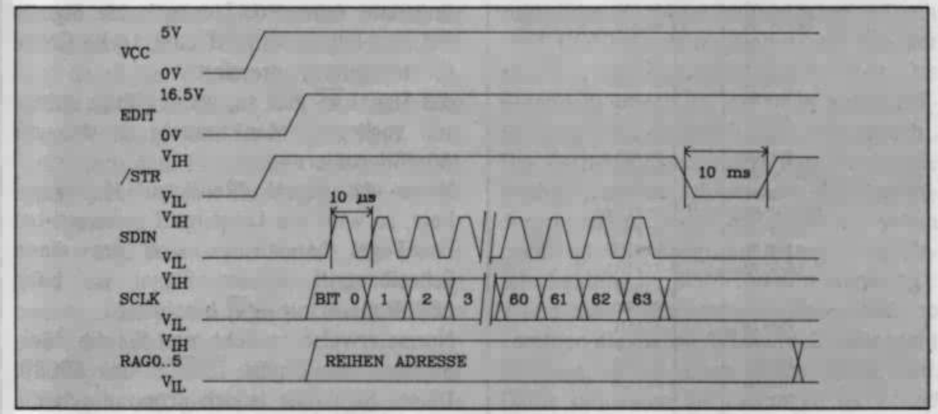


Bild 6. Der Signal-Zeitplan beim Programmieren

Zunächst erfolgt die Umsetzung der logischen Gleichungen in Daten für die einzelnen Reihen des Bausteins. Das kann z. B. von einem PAL-Assembler durchgeführt werden. Es ist aber sicherlich zum Verständnis der Möglichkeiten eines GALs besser, diese Schritte einmal von Hand auszuführen. In *Tabelle 6* ist das Ergebnis dieser Übersetzung am Beispiel des Adreßdekoders des Basic-Emuf (8-KByte-Version) dargestellt. Der Inhalt der Reihen 0...31 (Logikmatrix) wird durch die senkrechten Reihen der einzelnen Produktterme gebildet (siehe 1. Teil). Der Inhalt der Reihe 32 (UES) kann mit eigenen Daten gefüllt werden (z. B. Versionsnummer). Die Daten von Reihe 60 (ACW) werden in der Reihenfolge

PTD 0...31,
XOR (19, 18, 17, 16),
AC0
AC1 (19, 18, 17, 16, 15, 14, 13, 12),
XOR (15, 14, 13, 12)
PTD 32...63
gebildet.

Zur Programmierung wird das GAL in den „Edit"-Modus geschaltet (VCC ein, Vpp = 16,5 Volt ein, /STR=HIGH), danach wird der P/V-Anschluß auf „HIGH" gelegt (*Bild 6*).

Die Adresse der zu programmierenden Reihe wird auf die Anschlüsse RAG0...RAG5 (6 Bit) gelegt. Die Programmierdaten (64 Bit, bzw. 82 Bit für ACW) werden nun nacheinander in das Schieberegister mit der fallenden Flanke an SCLK (Zykluszeit min. 10 Mikrosekunden) eingetaktet (SDIN-Daten). Zur eigentlichen Programmierung wird auf den /STR-Anschluß ein 10 ms langer „LOW"-Impuls gelegt. Dieser Vorgang wird nun für alle Reihen durchgeführt. Danach wird die Programmierspannung (16,5 V) abgeschaltet und der Baustein kann in seiner Funktion getestet werden.

Zum Auslesen des Bausteins (im „Edit"-Modus) wird der P/V-Anschluß auf „LOW" gelegt. Die Adresse an den RAG0...RAG5-Anschlüssen bestimmt die Reihe zum Lesen. Der Inhalt wird mit einem kurzen /STR-Impuls (ca. 10 Mikrosekunden LOW) in das Schieberegister übernommen und mit der fallenden Flanke an SCLK am Anschluß SDOUT herausgetaktet (BIT0 zuerst).

Die Programmierung der Reihe 61 mit beliebigen Daten führt dazu, daß der Inhalt der Reihen 0...31 nicht mehr ausgelesen werden kann (Security Cell). Die Logikfunktion bleibt aber vollständig erhalten (siehe Bild 6).

Eigene Versuche mit diesen sehr interessanten Bauteilen seien empfohlen. □

Rolf-Dieter Klein

Z80-Schnittstelle für Amiga

Die Amiga-Computer gehören zu den Rechnern mit offener Architektur: Man kann sie erweitern, da alle internen Signale zugänglich sind. Der folgende Beitrag beschreibt, wie man einen NDR-Klein-Computer-Bus für die Erweiterung des Amiga nutzen kann: Eine preiswerte und trotzdem mächtige Lösung.

Die offene Systemarchitektur ist allen Amiga-Modellen eigen, dem Amiga 500 ebenso wie dem Amiga 1000 und 2000. Die folgende Beschreibung bezieht sich auf den Anschluß an den Amiga 500 und 1000, die einen sehr ähnlichen Erweiterungsbus besitzen: Der Amiga 1000 hat an der rechten Seite des Gehäuses eine Öffnung, hinter der sich dieser Stecker verbirgt. Beim Amiga 500 liegt dieser auf der linken Seite und ist gespiegelt.

Bild 1 zeigt die Steckerbelegung. Es handelt sich um einen 86poligen Stecker mit einem Pinabstand von 2,54 mm, man kann also handelsübliche Bauteile verwenden. Der Erweiterungsbus ist praktisch direkt mit dem Prozessor 68000 des Amiga verbunden, allerdings gibt es ein paar zusätzliche Signale, die man für die Schnittstelle braucht.

Die Signale

Hier sollen nur die wichtigsten Signale am Erweiterungsbus beschrieben werden, die genauen Zeiten findet man in jedem Handbuch zum 68000.

Die Leitungen A1...A23 sind die Adreßleitungen des Prozessors. Da der Amiga selbst einiges an Adreßraum verbraucht, sind natürlich nicht alle Adressen frei verwendbar. Anhand folgender Information können Sie aber einen freien Bereich aussuchen:

000000 256 KByte Hauptspeicher
040000 256 Kbyte Zusatzkarte beim Amiga 1000, beim Amiga 500 schon eingebaut
080000 darf nicht belegt werden
200000 freier Adreßraum für Erweiterungen
A00000 darf nicht belegt werden
BFD000 8520-B auf geraden Adressen
BFE001 8250-A auf ungeraden Adressen
C00000 für zukünftige Erweiterungen reserviert
DFF000 Spezial-Chips im Amiga
E00000 für zukünftige Erweiterungen reserviert
E80000 freier Adreßraum für Erweiterungen, wird aber auch für die automatische Konfiguration verwendet
F00000 darf nicht belegt werden, reserviert
F80000 System-ROM

Daraus ergeben sich zwei Bereiche, die für eigene Erweiterungen interessant sind: 200000...9FFFFF und E80000...EFFFFF.

Dabei ist noch zu beachten, daß viele Speichererweiterungen, die man meist intern einbauen kann, den Adreßraum ab 200000 belegen. Für die Busschnittstelle wurde daher der Bereich ab 800000 gewählt.

Da der Z80 zwischen Speicher und Ein-/Ausgabe unterscheidet, braucht man noch einen zweiten Bereich, den man am Besten bei E80000 einrichtet. Dieser Bereich ist jedoch nicht ganz unkritisch, denn ab Workbench 1.2 wird dort nach speziellen Bitmustern der Erweiterungsarchitektur gesucht. Werden diese aber nicht gefunden, so ist der Bereich anschließend frei. Daher sollte man dort in keinem Fall eine Speichererweiterung hinlegen.

Die Datenleitungen D0...D15 sind der 16-Bit-Datenbus. Wenn man Z80-Peripherie anschließen will, so braucht man allerdings nur acht Bit. Für die Schnittstelle werden D8...D15 verwendet, womit alle Erweiterungen auf geraden Adressen liegen. Wer seinen Speicher ausbauen will, der muß die Leitungen D0...D7 natürlich auch verwenden, er kann die Schnittstelle z.B. doppelt aufbauen und später jeweils eine andere Auswahlleitung ($\overline{LDS}$ bzw. $\overline{UDS}$) anschließen.

Zur Steuerung braucht man noch ein paar weitere Signale: $\overline{UDS}$ und $\overline{LDS}$ geben an, welche Bushälfte angesprochen wird: $\overline{UDS}$ steht für D8...D15, also für die geraden Speicheradressen, $\overline{LDS}$ für D0...D7, also für ungerade Adressen. Liegen beide Signale auf Low-Pegel, so wird eine 16-Bit-Größe (beide Hälften) adressiert.

das Signal $\overline{AS}$ gibt an, wann gültige Adressen vorliegen. Man braucht es also zur Adreßdecodierung.

Wenn das Signal PRNW auf High-Pegel liegt, so wird ein Lesezugriff gestartet, bei Low-Pegel handelt es sich um einen Schreibzugriff. Dieses Signal ist beim 68000 meist mit $R/\overline{W}$ beschriftet.

Normalerweise braucht man für die Rückmeldung das Signal $\overline{DTACK}$ des 68000. Dieses Signal ist jedoch schon durch die interne Logik des Amiga belegt und kann

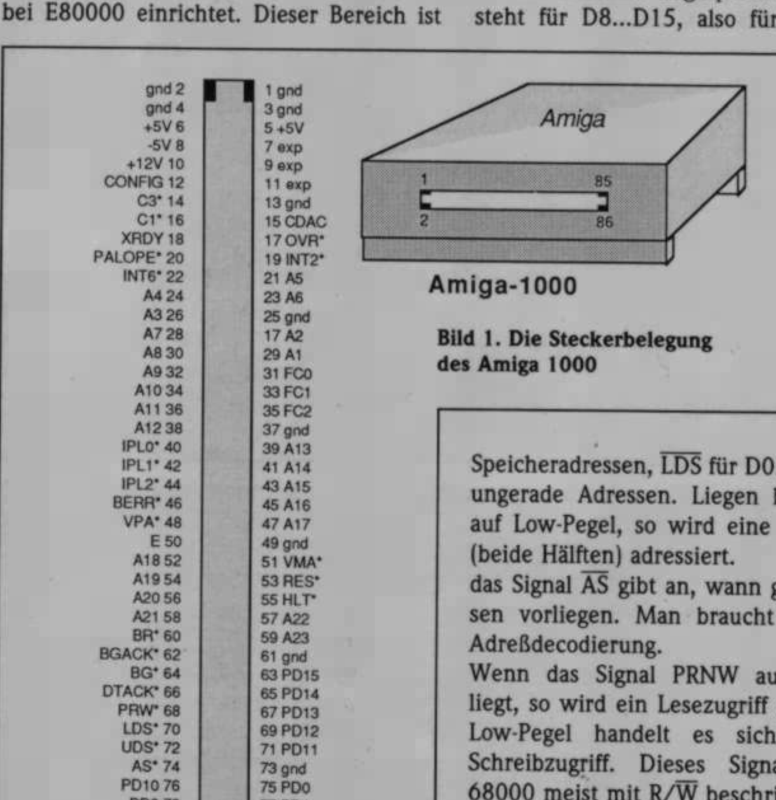


Bild 1. Die Steckerbelegung des Amiga 1000

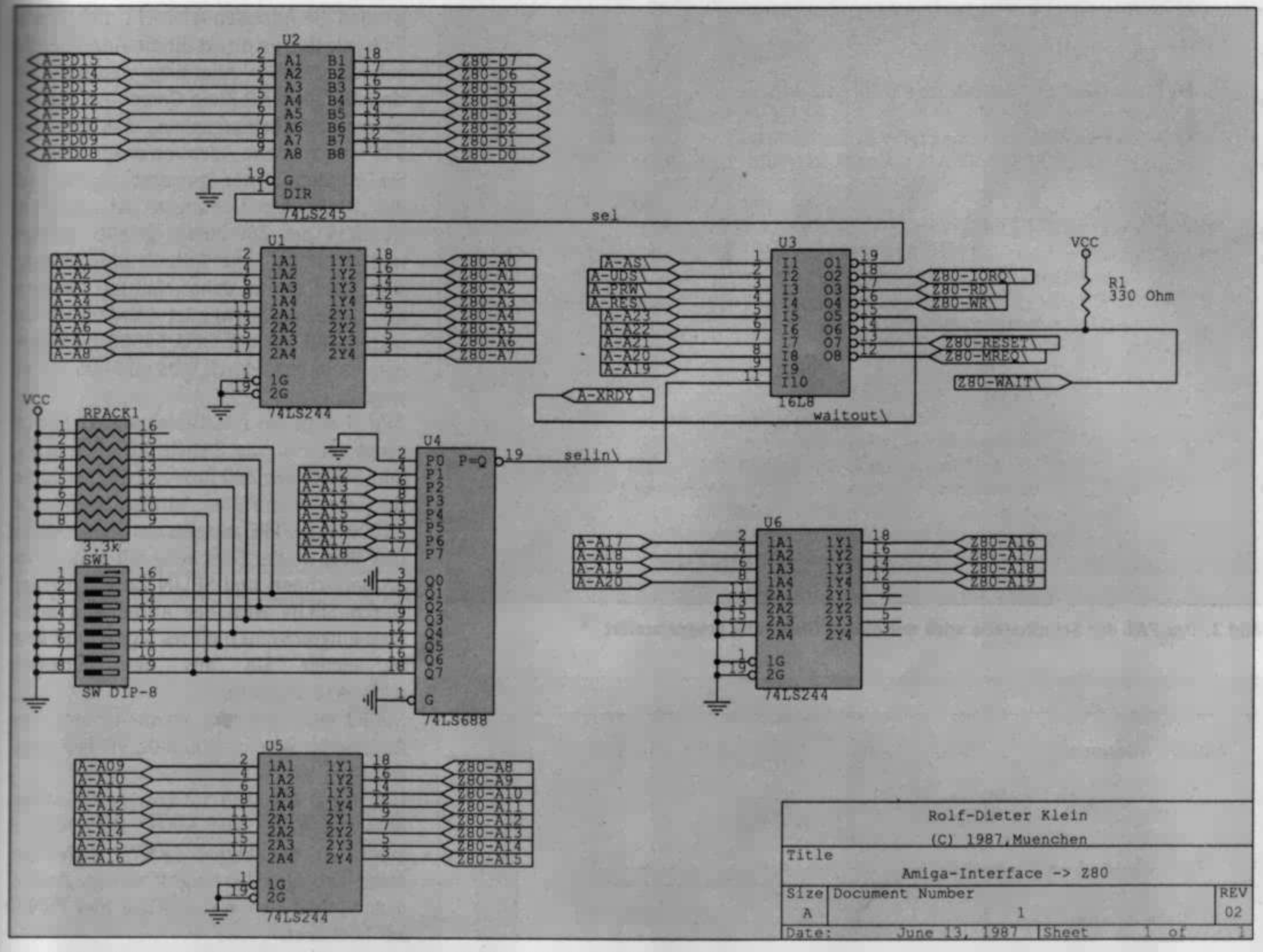


Bild 2. Das Schaltbild der Amiga-Z80-Busschnittstelle wurde mit OrCAD erstellt

nicht verwendet werden. Statt dessen steht der Eingang XRDY zur Verfügung: Legt man ihn auf Low, so wartet der 68000, bis man diesen Pegel wieder wegnimmt. Allerdings muß dazu das Low-Signal spätestens 60 ns nach der gültigen Adresse ($\overline{AS}$ auf Low) anstehen. Man kann dazu einen offenen Kollektor verwenden, da ein Pull-up-Widerstand schon im Amiga eingebaut ist.

Durch das Signal XRDY wird intern die Erzeugung des $\overline{DTACK}$-Signals verzögert und damit der Wartevorgang ausgelöst.

Bleibt noch das Signal $\overline{RES}$, das bei jedem Reset-Vorgang auf Low-Pegel liegt, also kurz nach dem Einschalten und bei Betätigung der Reset-Tasten (CTRL-Amiga-Amiga).

Das waren bereits alle Signale, die für die hier beschreibene Schnittstelle benötigt werden. Ein paar Amiga-spezifische Signale seien hier jedoch erwähnt:

Das Signal CONFIG ist für die automatische Konfiguration gedacht, die insbesondere bei der Workbench 1.2 verwendet wird: Es wird dazu von einer Erweiterung zur nächsten geleitet, der Adreßraum ist dabei frei programmierbar. Nur ein kleiner ROM-Bereich gibt dem Amiga Auskunft, welche Eigenschaften die jeweilige Hardware hat. So wird zunächst die erste in der Kette liegende Erweiterung angesprochen, da nur sie das Signal CONFIG erhält. Sie wird dann in einen vom Amiga-Betriebssystem zugewiesenen Adreßraum gelegt. Dann wird CONFIG weitergeleitet und von der nächsten Baugruppe ebenso behandelt. Das Besondere daran ist, daß man keine DIP-Schalter mehr auf den Baugruppen braucht. Das Betriebssystem liefert statt dessen die konfigurierbaren Adressen an das Anwenderprogramm, wenn dieses beim System nachfragt. Die Baugruppen tragen zur Unterscheidung im ROM-Teil eine Kennung.

Wenn man Interrupts verwenden will, so kommen die Signale IPL0...IPL2 dafür nicht in Frage, da sie schon intern belegt sind. Dafür kann man die Leitungen INT2 und INT6 anschließen.

Die Z80-Schnittstelle

Bild 2 zeigt die komplette Schaltung. Alle Signale sind über Treiber oder andere Bausteine geführt, so daß man am Z80-Bus ohne Sorgen um den Amiga arbeiten kann. Am Amiga-Stecker liegen auch +5 V, die man (beim Amiga 1000) mit maximal 1 A belasten kann. Man sollte damit nicht die ganze Erweiterung betreiben, speist man die Schnittstellen-Schaltung damit, so hat das den Vorteil, daß die Reihenfolge des Einschaltens keine Rolle spielt.

Alle Signale, die am Amiga-Stecker liegen, sind mit „A-" gekennzeichnet, alle Signale am Z80-Bus mit „Z80-".

In der Schaltung wird ein PAL 16L8 verwendet, um die Decodierung und Signalübersetzung durchzuführen. Die Programmierung sollte dank des mc-Universalprogrammierers kein Problem sein. Mit dem DIP-Schalter SW1 kann man den Ein-/Ausgabe-Adreßraum im Bereich E80000...EFFFFF genauer einstellen. Dabei

```
PAL16L8

Amiga Interface PAL Rolf-D.Klein 870614 r 1.2

NAS NUDS PRNW NRES A23 A22 A21 A20 A19 GND
NSELIN NMREQ NRESET NWAIT NWAITAUS NWR NRD NIORQ SEL VCC

IF (VCC) /SEL = NRD
IF (VCC) /NIORQ = /NAS * A23 * A22 * A21 * /A20 * A19 * /NSELIN
IF (VCC) /NRD = /NAS * /NUDS * PRNW * A23 * A22 * A21 * /A20 * A19 * /NSELIN
         + /NAS * /NUDS * PRNW * A23 * /A22 * /A21
IF (VCC) /NWR = /NAS * /NUDS * /PRNW * A23 * A22 * A21 * /A20 * A19 * /NSELIN
         + /NAS * /NUDS * /PRNW * A23 * /A22 * /A21
IF (/NWAIT) /NWAITAUS = /NWAIT
IF (VCC) /NRESET = /NRES
IF (VCC) /NMREQ = /NAS * A23 * /A22 * /A21

DESCRIPTION:    Dekodierung und Signalanpassung
                AMIGA-Expansion-Connector an Z80-Bus (NDR,ECB)
         IO-Adressbereich: E80000 to EFFFFF (512KByte), nur 256 benutzt.
         Speicherbereich  : 800000 to 9FFFFF (2MByte)
         Die Adressierung erfolgt nur auf geraden Adressen.
         Damit ist der Speicherbereich auf 1MByte begrenzt.
```

Bild 3. Das PAL der Schnittstelle wird mit dieser Gleichung programmiert

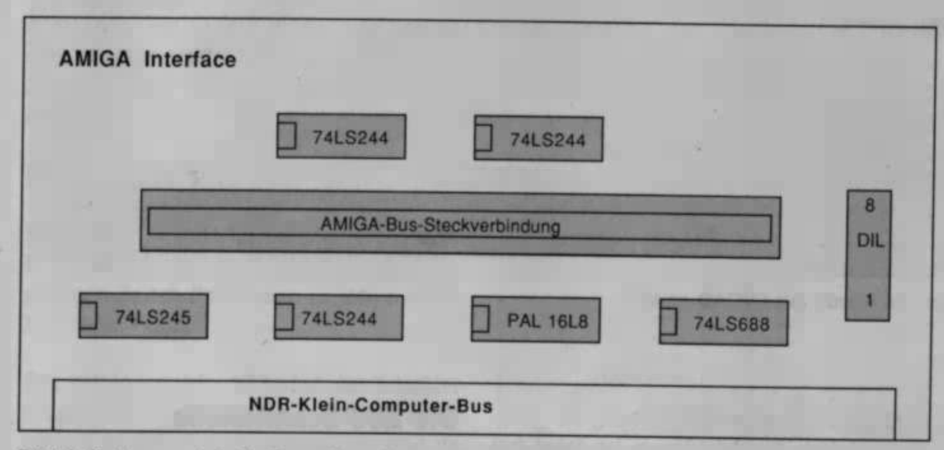


Bild 4. So lassen sich die Bauteile auf einer Europakarte anordnen

Item	Quantity	Reference	Part
1	3	U1,U5,U6	74LS244
2	1	U2	74LS245
3	1	U3	16L8
4	1	U4	74LS688
5	1	SW1	SW DIP-8
6	1	RPACK1	3.3k
7	1	R1	330 Ohm

Bild 5. Die Bauteile-Liste

werden die Adressen A1...A11 nicht beeinflußt; sie dienen direkt für die Adressierung auf dem Z80-Bus. Allerdings wertet die Peripherie des NDR-Klein-Computers nur die unteren acht Leitungen aus, man kann also 256 verschiedene Adressen ansprechen.

Sie werden sicher gemerkt haben, daß die Amiga-Adreßleitungen A1...A20 an A0...A19 des Z80-Busses geleitet werden und damit um eine Adreßleitung versetzt sind. Dies kommt daher, daß nur die geraden Adressen verwendet werden, da die Leitung A0 bei der CPU 68000 nicht vorhanden ist bzw. durch $\overline{\text{UDS}}$ und $\overline{\text{LDS}}$ ersetzt wird.

Bild 3 zeigt die PAL-Gleichung. Damit erzeugt der PAL die Signale $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{IORQ}}$ und $\overline{\text{MREQ}}$ des Z80-Busses. $\overline{\text{IORQ}}$ wird aktiv (Low), wenn eine Adresse im Bereich E80000...EFFFFF ansteht, das Signal $\overline{\text{AS}}$ auf Low liegt (in der Gleichung mit NAS = Not AS bezeichnet) und $\overline{\text{SELIN}}$ aktiv (Low) ist. Durch $\overline{\text{SELIN}}$ wird der Adreßraum nochmals eingeschränkt; dieses Signal wird vom Vergleicher U4 aus den Adressen A12...A18 abgeleitet.

$\overline{\text{MREQ}}$ wird aktiviert, wenn $\overline{\text{AS}}$ und eine Adresse im Bereich 800000...9FFFFF angesprochen wird.

Das Signal $\overline{\text{RD}}$ wird auf Low gelegt, wenn entweder $\overline{\text{IORQ}}$ oder $\overline{\text{MREQ}}$ aktiv ist und zusätzlich das Signal PRNW High-Pegel zeigt, also eine Lesezugriff erfolgt. Analog dazu verhält sich $\overline{\text{WR}}$, nur daß hier PRNW auf Low liegt.

Durch das Signal $\overline{\text{SEL}}$ am Ausgang Pin 19 des PAL schaltet den Datenbustreiber U2 bei einem Lesezugriff in Richtung Amiga um. Das $\overline{\text{WAIT}}$-Signal des Z80 wird praktisch direkt zum Amiga-Eingang XRDY geleitet, wobei der PAL-Baustein so programmiert wurde, daß ein offener Kollektor an Pin 15 entsteht. Die Formulierung

IF (/NWAIT) /NWAITAUS = /NWAIT

bewirkt nämlich, daß der Tri-State-Treiber des PAL nur dann eingeschaltet wird, wenn $\overline{\text{WAIT}}$ auf Low liegt – im Ergebnis also ein offener Kollektor. Das $\overline{\text{RESET}}$-Signal wird direkt weitergeleitet, hier dient der PAL nur als Treiber.

Bild 4 zeigt einen möglichen Aufbau auf einer Europakarte. Die Karte wird dann z. B. auf den NDR-Klein-Computer-Bus gesteckt oder über ein (nicht zu langes !) Flachbandkabel an einen EC-Bus angeschlossen. *Bild 5* zeigt die Bauteile-Liste.

Testsoftware

Ein paar Beispielprogramme sollen den Zugriff auf den Z80-Bus verdeutlichen. *Bild 6*

```
#include <stdio.h>

main()
{
 int i,j;
 long k;
 char *ptr;
 printf("Dump Utility  Rolf-Dieter Klein\n");
 printf("Startadresse :");
 scanf("%lx",&ptr);
 printf("\n");
 for (j=0 ; j<16; j++) {
  printf("%06lx : ",ptr);
   for (i=0 ; i<16; i++) {
    k = (long *) (*ptr);
    printf(" %02lx",k & 0xff);
    ptr++;
   }
  printf("\n");
 }
}
```

Bild 6. Auch für andere Probleme: Ein Speicher-Dump-Programm

```
#include <stdio.h>

main()
{
 int i,j;
 long k;
 char *ptr;
 printf("IO-Dump Utility  Rolf-Dieter Klein\n");
 printf("Startadresse :");
 scanf("%lx",&ptr);
 printf("\n");
 for (j=0 ; j<16; j++) {
  printf("%06lx : ",ptr);
   for (i=0 ; i<16; i++) {
    k = (long *) (*ptr);
    printf(" %02lx",k & 0xff);
    ptr++; ptr++; /* da ios auf even adr. liegen */
   }
  printf("\n");
 }
}
```

Bild 7. Dieses Programm zeigt nur die geraden Adressen

```
#include <stdio.h>

main()
{
 int i,j;
 long k;
 char *ptr,*erg;
 ptr = 0xe80000; /* Basis IO-Adresse */
 printf("LED-Schalter Test\n");
 for (j=0; j<200; j++) {
  *(ptr + 0x30 * 2) = *(ptr + 0x30 * 2);
  *(ptr + 0x31 * 2) = *(ptr + 0x31 * 2);
 printf("test laeuft..\n");
 }
}
```

Bild 8. Ein kleiner Test der IOE-Baugruppe

```
#include <stdio.h>

main()
{
 int i,j;
 long k;
 char *ptr,*erg;
 ptr = 0xe80000; /* Basis IO-Adresse */
 printf("ADC0816 Test\n");
 for (j=0; j<200; j++) {
  for (i=0; i<4; i++) {
   erg = ptr +  0xe0*2 +  i*2;
   *erg = 0;
   printf("adc %d=",i);
   erg = ptr +  (0xe0*2 + 2);
   printf("%3d ",(*erg) & 0xff);
  }
  printf("\n");
 }
}
```

Bild 9. So liest der Amiga Werte aus dem ADC0816

zeigt ein C-Programm, mit dem man sich beliebige Speicherbereiche des Amiga als Hexdump ausgeben lassen kann. Dabei werden gerade und ungerade Adressen angesprochen.

Bild 7 zeigt ein Programm, mit dem man nur gerade Adressen anspricht. Bei beiden Programmen wird nach der Startadresse gefragt, die hexadezimal eingegeben werden muß. Es werden dann 16 Zeilen mit je 16 Byte ab der Startadresse angezeigt.

Das in *Bild 8* abgebildete Programm spricht die IOE-Baugruppe an: Der Wert von Port 30h wird eingelesen und auf der Adresse 30h wieder ausgegeben, entsprechend wird mit Port 31h verfahren. Als einfachen Test kann man am Eingabeport 30h einen DIP-Schalter und am Ausgabeport LEDs anschließen: Die LEDs müssen sich mit den DIP-Schaltern ein- und ausschalten lassen. Im Programm werden die Portadressen immer mit zwei multipliziert, da ja nur gerade Adressen vorkommen – darauf sollte man bei der Adressierung immer achten.

Bild 9 schließlich zeigt ein A/D-Wandler-Programm, das die Baugruppe ADC0816 des NDR-Klein-Computers verwendet. Es werden dabei die ersten beiden Kanäle als Zahlenwerte in einer Reihe 200mal auf dem Bildschirm ausgegeben. Den Programmlauf kann man übrigens jederzeit durch CTRL-C abbrechen.

Beim Betrieb mit dem ADC0816 muß man eigentlich warten, bis eine Wandlung abgeschlossen ist, doch wird in diesem Programm durch das „printf" nach der Startanweisung (*erg = 0) ausreichend Zeit verbraucht (mehr als 100 µs). So ist es auch leichter möglich, Schaltungsfehler der Schnittstelle einzukreisen.

Literatur

[1] Amiga Hardware Reference Manual, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986

[2] Hardware-Developers preleminary: „Amiga Expansion Architecture", Commodore, 4. Feb. 1986

Ralf Aron

Die mc-ST-Tastatur

Neue Tastatur für den Atari ST selbstgebaut

Viele Anwender haben sich bestimmt schon über das äußere Erscheinungsbild des Atari ST geärgert. Die Original-Tastatur läßt es nicht zu, den Rechner als Textverarbeitungssystem zu benutzen. Deshalb stellen wir eine bessere Tastatur vor, die auch Schnellschreibern genügen wird.

Der Atari ST ist von Seiten der Software und der technischen Daten hervorragend als Textverarbeitungs-Computer geeignet. Neben der unergonomischen Tastatur ist ein weiteres Manko das Gehäusekonzept, das Hardware-Erweiterungen fast unmöglich macht.

Da blickt man schon neidisch zu den PC-Besitzern herüber und träumt von einem Atari ST, der nicht wie ein Rechner-Puzzle (hier ein Laufwerk, da ein Netzteil....) aussieht und nicht von einer Spielzeugtastatur bedient wird.

Es spricht also einiges dafür, dem Atari ST ein gescheites Äußeres zu geben. Die neuen Rechner der Mega ST-Serie zeigen, daß man den Atari ST auch schöner verpacken und mit einer flexiblen, frei beweglichen Tastatur ausrüsten kann.

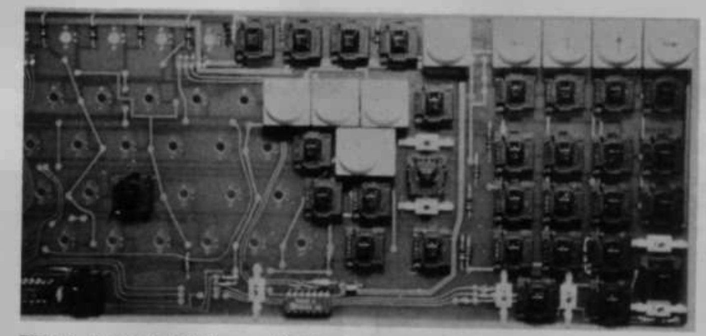


Bild 1. Ausschnitt der mc-ST-Tastatur, zum Teil bestückt

Der Atari ST gehört heute wohl mit zu den meist gekauften Rechnern. Dies liegt eindeutig in seinem guten Preis/Leistungsverhältnis begründet. Bekommt man doch bereits für ca. 2000 DM eine komplette Rechneranlage mit Drucker und allem was dazu gehört. Seine größten Pluspunkte sind, meiner Meinung nach, die Schwarzweiß-Grafik, die ihresgleichen sucht und der 68000-Mikroprozessor.

Ein kaum noch zu überblickendes Softwareangebot machen den Atari ST noch attraktiver und wer keine Lust auf TOS, GEM und Maus hat, programmiert halt unter RTOS oder OS/9 oder unter einem anderen Betriebssystem wie z.B. Eumel oder CP/M. Trotzdem ist der Atari ST (noch) nicht das Maß aller Dinge.

Spätestens wenn man seitenlange Listings abtippt oder der Gedanke einer Hardware-Erweiterung keimt, merkt man, warum der Computer so billig ist. Das alles sind aber nur Schönheitsfehler, die sich beheben lassen.

Es gilt also, dem Atari ST ein neues Gehäuse zu verschaffen. Als Vorbild hierfür steht natürlich das Gehäusekonzept der XT/AT-Rechner mit abgesetzter Tastatur und einem großen Gehäuse, das alle „lebenswichtigen" Teile enthält und genügend Platz für Zusatz-Hardware bietet. Ein besonders interessanter Beweggrund für einen Umbau ist der Anschluß eines Bus-Systems an den Atari ST, um das reichhaltige Angebot an Peripherie-Bausteinen zu nutzen. Es gibt bereits eine große Auswahl an Bus-Systemen. Mir bekannt sind der ECB-, IBM-, IEC-, VME- und der mc-68000-Bus.

Besonders interessant ist dabei das Bus-System der Firma Rhothron, da dieses speziell auf den Atari ST zugeschnitten ist. Außerdem ist dieses System voll kompatibel zum mc-68000-Bus. Hiermit könnten Erweiterungskarten entwickelt werden, die nicht nur für einen Rechnertyp geeignet wären. Es ist also durchaus denkbar, daß auf diesem Gebiet noch Interessantes passiert. Dieses wird jedoch erst möglich wenn die Mutterplatine sich in einem geräumigen Gehäuse befindet, das auch den Bus aufnehmen kann.

Was mich aber am meisten wurmte war die schreckliche Tastatur, die allen ergonomischen Anforderungen Hohn spricht. An der neuen Mega ST-Serie ist erkennbar, daß selbst Atari mit dem alten Konzept nicht zufrieden ist.

Bevor ich ins Detail gehe, möchte ich ein paar warnende Worte sagen. Falls Sie noch nie mit einem Lötkolben gearbeitet haben und Elektronik ein Fremdwort für Sie ist, so sollten Sie Ihren Atari ST lieber so lassen, wie er jetzt ist. Ohne Löten und Nachmessen geht es nicht. Falls Sie den Umbau vornehmen wollen, ist es unbedingt nötig, zumindest ein Vielfach-Meßgerät zu haben. Ein Oszilloskop kann auch nicht schaden, da es bei der Fehlersuche sehr nützlich ist. Ein weiteres Hilfsmittel sind die Schaltpläne des Atari ST, die bei den meisten Händlern zu bekommen sind.

Die neue Tastatur

Zuerst möchte ich die Tastatur vorstellen, da sie auch ohne große Umbauarbeiten an den Atari ST angeschlossen werden kann. Dies ist besonders für all diejenigen interessant, die nur die neue Tastatur anschließen wollen und sonst keine weiteren Umbau-Ambitionen haben. Das Vorbild der neuen Atari ST-Tastatur ist die große DIN-Tastatur von Elzet-80. Die Tastatur fällt besonders durch ihre Kompaktheit auf. Die Abmessungen betragen 400 * 140 mm. Dadurch ist gewährleistet das alle Tasten gut zu erreichen sind (*Bild 1*).

Die gesamte Tastatur umfaßt 96 Tasten, davon sind allerdings nur 95 belegt (*Bild 2*). Aufgeteilt ist die mc-ST-Tastatur in die vier Blöcke Haupt- und Funktionstastenfeld (16 Tasten), Zahlen- (14 Tasten) und Cursor-Block (8 Tasten). Die Tasten werden von Siemens gefertigt. Die Tastenbeschriftung entspricht weitgehend der des Atari ST.

Welche Tasten eventuell abweichen, läßt sich zu diesem Zeitpunkt, wo der Artikel entstanden ist, noch nicht sagen.

Zur Ergonomie ist noch zu sagen, daß der Anschlag der Tasten angenehm hart und der Druckpunkt deutlich spürbar ist. Dadurch wird ein deutlich besseres Tippgefühl als auf der Original-Tastatur erzeugt. Durch den harten Anschlag werden auch die Fehlbedienungen durch versehentliches Berühren von Tasten vermindert.

Die Tastatur wird über ein fünfadriges Kabel mit der Mutterplatine verbunden.

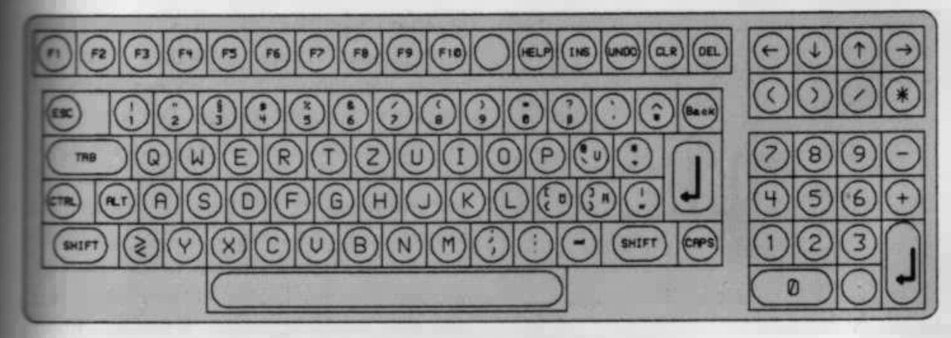


Bild 2. So sind sie Tasten auf der mc-ST-Tastatur angeordnet

Maus- und Joystick-Anschluß befinden sich hinten am Fuß des Tastaturgehäuses. Die Anschlüsse für Maus und Joystick auf der Mutterplatine werden nicht mehr benötigt. Jedem, der schon mal seinen Atari ST aufgeschraubt hat, wird aufgefallen sein, daß der Tastatur-Prozessor nicht eingelötet wurde, sondern in einer Fassung steckt. Dies brachte mich auf die Idee, den Schaltplan der Originaltastatur zu entschlüsseln und anhand dieses Planes eine Platine für die neue Tastatur zu entwerfen. Die einzigen Änderungen vom Original sind die Einfügung eines Puffer-ICs (7407) für die seriellen Datenleitungen von und zur Mutterplatine und eine veränderte Pinbelegung des Bustreiber-ICs (74LS244). Die veränderte Pinbelegung war notwendig, um die Leiterbahnführung zu vereinfachen (*Bild 3*).

Es ist also gewährleistet, daß die Tastatur voll kompatibel zum Original ist und ohne Anpassungssoftware funktioniert. Außerdem treten durch dieses Konzept keine Probleme mit Anwendungs-Programmen auf. Um es nochmal deutlich zu machen, Sie benötigen für den Betrieb der mc-ST-Tastatur den Tastatur-Prozessor des Atari ST.

Diese Beschreibung bezieht sich auf die Atari-Typen 520 ST+, 260 ST und 520 STM. Der Atari 1040 ST weist ein anderes Platinenlayout auf und benötigt deshalb eine andere Tastatur-Schnittstelle. Die Anschlüsse für Maus und Joystick befinden sich, im Gegensatz zu den anderen Modellen, auf der Tastaturplatine. Aus diesem Grunde besteht das Verbindungskabel zur Mutterplatine beim Atari 1040 ST nur aus sieben Leitungen und nicht 17, wie bei den anderen Typen (*Bild 4*). Soweit es mir möglich ist, werde ich die Unterschiede des Atari 1040 ST berücksichtigen und Änderungsvorschläge geben.

Ein kompletter Bausatz, bestehend aus Platine, Tasten, Tastenkappen, Gehäuse und allen sonstigen elektronischen Bauteilen (natürlich ohne Tastaturprozessor) ist beim Elektronikladen in Detmold erhältlich.

Die erste Phase

Löten Sie alle passiven Bauteile in die Platine ein. Das sind die Widerstände R12, R13 (680 Ω) und R14 bis R36 (10 kΩ), die Dioden D1 bis D15 (achten sie hierbei bitte besonders auf die Polarität der Dioden), die Kondensatoren C2 bis C6 und den 4-MHz-Quarz. Da sich das Bohrschema für die Tasten mehrfach geändert hat, sollten Sie einige Dinge beachten. Die Widerstände R15 bis R26 müssen, nachdem sie eingelötet sind, ein ganz kleines Stück zur Seite gebogen werden, damit die Tasten richtig eingesetzt werden können. Beim Widerstand R27 müssen die Anschlußbeinchen etwas länger ausfallen, damit er ausrei-

Bild 3. Der Schaltplan für die mc-ST-Tastatur

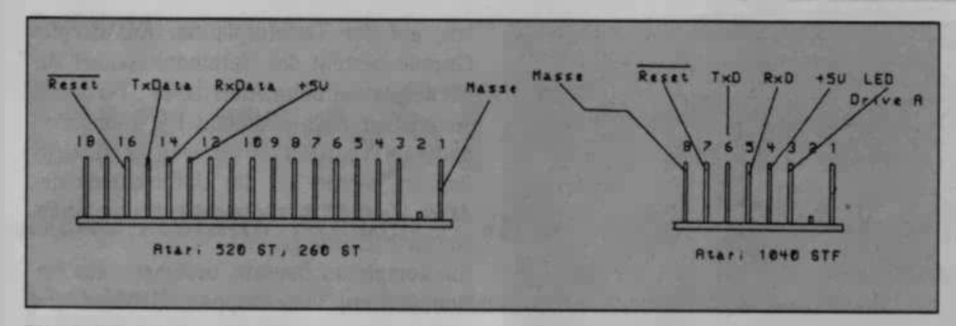


Bild 4. Die Schnittstelle zwischen Atari ST und der mc-ST-Tastatur

chend zur Seite gebogen werden kann. Leider ist es aus Kostengründen nicht vertretbar, diese kleinen Schönheitsfehler zu beheben.

Als nächstes löten Sie die Fassungen für den Tastaturprozessor (IC1) und den Buffer (IC3) ein. Der Bustreiber (IC2) muß direkt eingelötet werden, da er sich direkt unter der Leertaste befindet und daher nicht genug Platz für eine Fassung vorhanden ist (*Bild 5*). Achten Sie deshalb besonders darauf, daß Sie das IC richtig herum einlöten. Jetzt müssen Sie noch die Widerstände R1 bis R11 und den Kondensator C1 (100 µF) einlöten. Die Widerstände sind auf der Lötseite bei den Anschlußlötaugen für Maus, Joystick und Verbindungskabel als Widerstandsnetzwerk einzulöten (*Bild 6*). Die Anschlußlötaugen befinden sich in Höhe des Gehäusefußes. Jedes Anschlußlötauge hat zwei Lötlöcher. In den unteren Lötlöchern werden die oben besprochenen Widerstände eingelötet. Die oberen Lötlöcher dienen dazu, einreihige Pfostenstecker aufzunehmen, die natürlich ebenfalls von der Lötseite her eingelötet werden. Über Buchsen können dann die entsprechenden Verbindungen zu den Anschlußsteckern für Maus und Joystick hergestellt werden, ebenso das Verbindungskabel zur Mutterplatine. Löten Sie bitte noch keine Tasten ein, dazu kommen wir später.

Bevor wir uns ganz dem Gehäuse widmen, überprüfen Sie nochmal die soweit aufgebaute Platine auf Kurzschlüsse, kalte Lötstellen, Polarität der Dioden und des Kondensators und ob alle Bauteile eingesetzt worden sind.

Die zweite Phase

In den Fuß des Tastaturgehäuses müssen jetzt noch die Ausschnitte für die 9-poligen Sub-D-Stecker (Maus und Joystick) angefertigt werden, wobei es am persönlichen Geschmack und Bedarf liegt, ob nur der Maus-Anschluß oder beide Anschlüsse nach außen geführt werden. Im Fuß des Gehäuses befinden sich bereits zwei Löcher. Das größere von beiden (das Rechte von hinten gesehen) benutzen Sie als Kabeldurchführung für das Verbindungskabel. Links und rechts von diesem Loch können dann die Stecker angebracht werden. Das Ausschneiden der Löcher ist eine etwas heikle Angelegenheit, da nicht sehr viel Platz zum bohren und feilen ist. Nehmen Sie sich also ein bißchen Zeit und Geduld für das Ausschneiden der Löcher. Ist diese Arbeit erledigt, werden die Sub-D-Stecker eingesetzt und mit den entsprechenden 8-poligen Buchsenleisten, die nachher auf die Tastaturplatine aufgesteckt werden, verdrahtet. Die Anschlußbelegung der Sub-D-Stecker ist in *Bild 7* dargestellt.

Als nächstes wird das Verbindungskabel angebracht. Ein Knoten direkt hinter der Kabeldurchführung dient als Zugentlastung. Löten Sie dann eine 5-polige Buchsenleiste an das Kabel. Das Verbindungskabel sollte abgeschirmt sein.

An das andere Ende des Verbindungskabels wird nun eine einreihige, 18-polige Buchsenleiste angelötet, die nachher auf die Stiftleiste der Mutterplatine gesteckt wird. Besitzer eines Atari 1040 STF benötigen eine 8-polige Buchsenleiste. Eine weitere Besonderheit beim Atari 1040 STF ist der Pin 3. An diesem Pin liegt das Signal für die Laufwerksdiode an, die anzeigt, wann auf das Laufwerk zugegriffen wird. Die Anschlußbelegung der Stiftleiste ist in *Bild 4* dargestellt. Die restlichen Pins der Stiftleiste werden nicht benötigt (auch bei den Atari 520 ST-Modellen reichen demnach fünf Leitungen aus).

Bevor es weitergeht, werden nochmal alle Stecker und Buchsen daraufhin kontrolliert, ob sie richtig beschaltet sind.

Die dritte Phase

Jetzt geht es dem Atari ST an den Kragen. Lösen Sie die sechs Schrauben auf der Unterseite und entfernen dann den oberen Teil des Gehäuses. Um die Tastatur abzunehmen, braucht nur noch die Tastaturbuchse abgezogen zu werden. Der Tastaturprozessor 6301 befindet sich auf der Tastaturplatine. Mit einem mittelgroßen Schraubenzieher kann dieser vorsichtig aus der Fassung gehebelt werden. Verwahren Sie das teure und einzige Stück bitte an einem sicheren Ort. Bauen Sie nun die neue Tastatur soweit zusammen, daß ein erster Test vorgenommen werden kann.

Bei allen weiteren Aktionen ist zu bedenken, daß der Tastaturprozessor ein einmaliges Exemplar ist und im Handel nicht nachgekauft werden kann! Deshalb nochmal die Aufforderung: Überprüfen Sie alle Steckverbindungen auf richtige Beschaltung und Polarität.

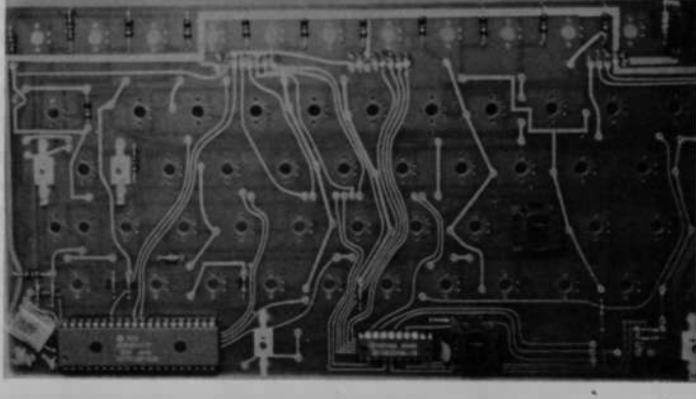


Bild 5. Auf diesem Ausschnitt sieht man den Tastatur-Prozessor und den Treiber

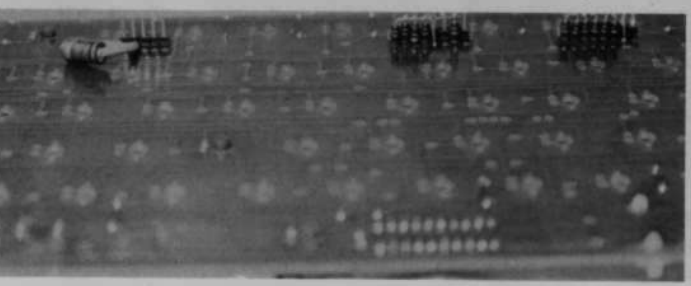


Bild 6. So müssen die Widerstände und der Kondensator eingelötet werden

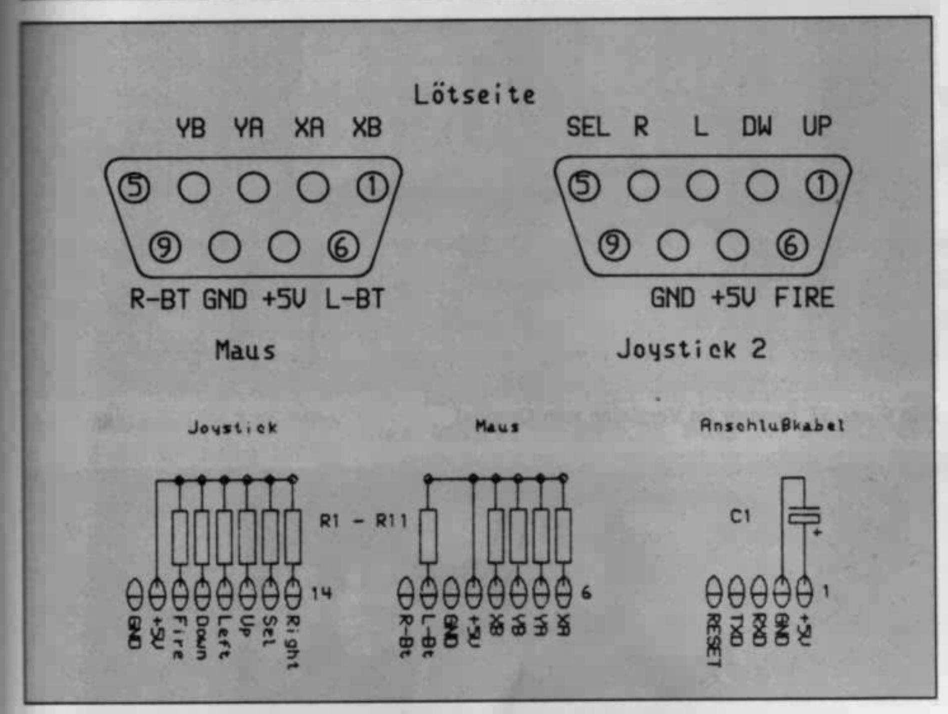


Bild 7. Die Anschlußbelegung der Sub-D-Stecker für Maus und Joystick

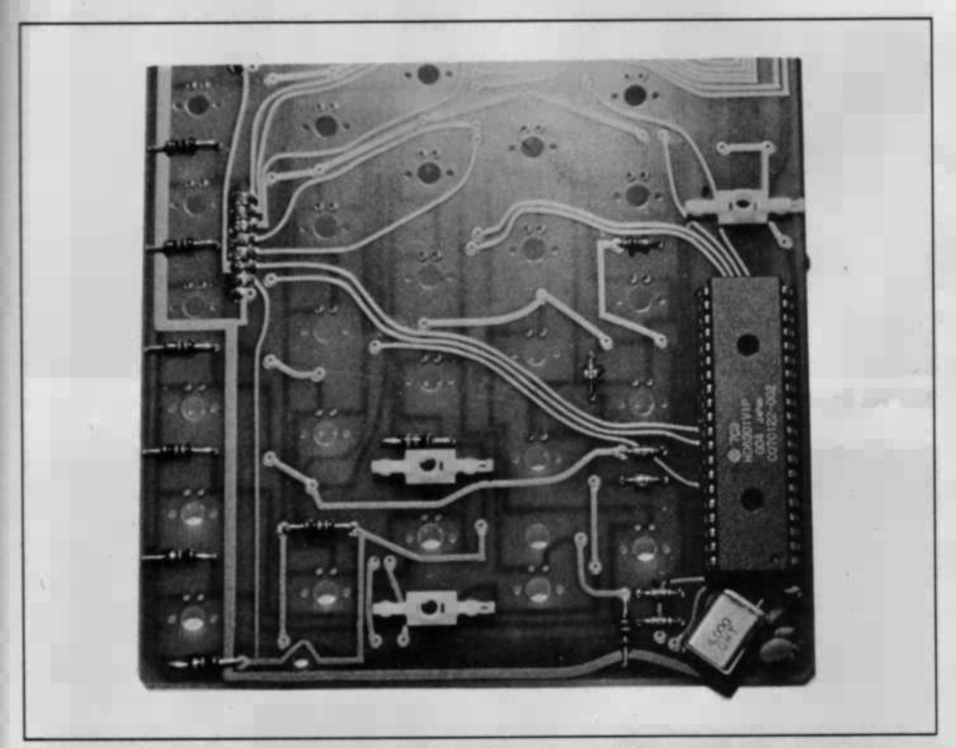


Bild 8. Der Tastatur-Prozessor des Atari ST muß in die mc-ST-Tastatur eingesetzt werden

Mit einem Meßgerät wird nun ein eventueller Kurzschluß zwischen Versorgungsspannung und Masse (Widerstand messen) aufgespürt (Anschlußpins 1 und 2). Es sollte ein Widerstand von größer 140 kΩ angezeigt werden. Ist alles ok, kann jetzt das IC3 (den Prozessor bitte noch nicht) eingesteckt werden. Verbinden Sie nun die Tastatur mit dem Atari ST und schalten diesen unter starkem Herzklopfen ein. Der Rechner müßte ganz normal den gewohnten Desktop zum Vorschein bringen. Jetzt wird mit einem Voltmeter die Versorgungsspannung an der Fassung des Prozessors und den anderen ICs gemessen. Wo und wie die Spannung anliegen muß, ist dem Schaltplan zu entnehmen; sie sollte bei ungefähr fünf Volt liegen. Berühren Sie die Bauteile ruhig auch mit einem Finger (Temperaturkontrolle).

Ist alles zur Zufriedenheit verlaufen, schalten Sie den Rechner wieder aus. Nach einer kleinen Wartezeit kann nun der Tastaturprozessor in die Fassung eingesetzt werden (*Bild 8*). Kontrollieren Sie 20mal, ob das IC auch wirklich richtig herum eingesetzt worden ist. Nachdem die Maus angeschlossen ist, wird der Rechner wieder eingeschalten. Der Mauscursor sollte sich wie gewohnt bewegen lassen. Versuchen Sie auch, eine Datei zu öffnen und ein Programm zu starten (am besten einen Editor). Durch Überbrücken von Schalterkontakten können Sie nun testen, ob sich Buchstaben eingeben lassen. Ist auch dieser Test erfolgreich verlaufen dann steht der Vollendung nichts mehr im Wege.

Die letzte Phase

Nachdem der Rechner wieder ausgeschaltet wurde, wird die Tastaturplatine mit den noch fehlenden Teilen bestückt. Zuerst werden die Tasten eingelötet. Gehen Sie dabei vorsichtig zu Werke, damit die kleinen Anschlußbeinchen unter den Tasten nicht abbrechen. Als nächstes werden die Führungsmechaniken für die Tasten TAB, RETURN, ENTER, BLANK und 0 (Zahlenblock) eingesetzt. Jetzt müssen noch die Tastenkappen aufgesetzt werden. Aufschluß über die Lage der einzelnen Tasten gibt Bild 2. Bevor alles wieder richtig verkabelt wird, führen Sie noch einmal eine gründliche Sichtkontrolle durch. Überprüfen Sie die Schaltung an dieser Stelle nochmal auf einen Kurzschluß zwischen Masse und der Versorgungsspannung. Als letzte Tat wird noch der Tastaturdeckel aufgesetzt und an den Seiten mit zwei Schrauben fixiert.

Bei einem Fehler ...

Sollte die Tastatur unerwarteter Weise nicht funktionieren, überprüfen Sie folgende Punkte:

1. Überprüfen Sie alle Leiterbahnen mit einem Widerstandsmeßgerät auf Haarrisse.

2. Prüfen Sie, ob Sie alle Bauteile eingesetzt und sauber verlötet haben (kalte Lötstellen, Kurzschlüsse).

3. Haben Sie alle Verbindungen richtig verlötet und aufgesteckt?

4. Betrachten Sie die Versorgungsspannung bei eingeschaltetem Rechner mit einem Oszilloskop. Sie muß frei von Störungen sein.

5. Am Pin 2 des Tastaturprozessors muß ein sinusförmiges Signal mit einer Frequenz von 4 MHz anliegen. Der Tastaturprozessor muß dazu natürlich eingesteckt sein.

6. Klemmen Sie das Oszilloskop an die RxD-Leitung und bewegen Sie die Maus. Es müßte ein deutliches Signal zu erkennen sein. Wenn Sie eine Taste drücken, sollte nur ein kurzer Impuls zu sehen sein. Ist dies nicht der Fall, wiederholen sie die Messung vor dem IC3 (Pin 11) und direkt am Tastaturprozessor an Pin 12. Die Reset- und TxD-Leitungen müssen im Normalzustand auf High-Pegel liegen (ca. 5V). Beim Einschalten des Rechners oder beim Drükken des Resetknopfes muß die Reset-Leitung kurz auf Low-Pegel gehen und auf der TxD-Leitung eine kurze Impulsfolge erkennbar sein. An Pin 8 des Prozessors müssen kurze Impulse zu sehen sein (ca. 500 Hz). Diese Impulse dienen dazu, das IC2 zu aktivieren.

Normalerweise sollte es keine Probleme geben, da die Elektronik der Tastatur nicht sehr komplex ist. Selbst in der Entwicklungsphase gab es bei mir keine Schwierigkeiten und gleich der erste Platinenentwurf hat funktioniert. □

Stückliste für die Tastatur

- 34 × 10 kΩ
- 2 × 680 Ω
- 15 × 1N4148
- 1 × 7407
- 1 × 74LS244
- 1 × 4 MHz Quarz
- 1 × 100 µF/25 V
- 2 × 6,8 pF Keramik
- 2 × 100 pF Keramik
- 1 × 100 nF Keramik
- 1 × 40polig gedrehte Fassung
- 1 × 14polig gedrehte Fassung
- einreihige Pfostenstecker und die dazu passende einreihige Buchsenleiste
- 2 × 9polige Einbau Sub-D-Stecker
- 2–3 Meter 5adriges, abgeschirmtes Kabel
- 96 × Siemens-Tasten und der dazugehörige Kappensatz
- 1 × Gehäuse



Bild 9. mc-ST-Tastatur im Vergleich zum Original

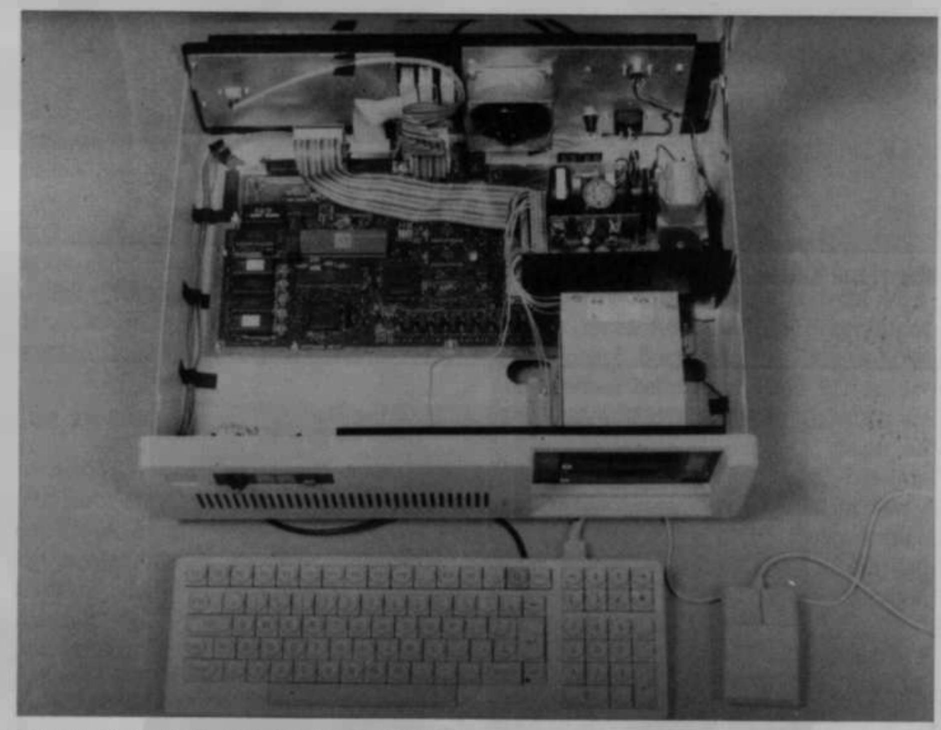


Bild 10. So sieht ein umgebauter Atari ST aus

Bild 11. Umgebauter Atari ST mit der mc-ST-Tastatur

Dieter Strauß

80386-Power von Osborne

Mit dem Modell O6T-386 ist Osborne in die Arena der 386er-Computer eingetreten. Der Preis des Rechners liegt in einem Bereich, in dem sich vor nicht allzu langer Zeit zahlreiche AT-kompatible Computer tummelten. Hält ein preiswerter 386er-Computer, was er verspricht, oder tut's auch ein AT?

Um die Leistungsfähigkeit des Modells O6T-386 zu beurteilen, haben wir das System in der Konfiguration mit 40-MByte-Festplatte, 1,2-MByte-Floppy, Hercules-kompatibler Grafikkarte und 2,56 MByte RAM auf Herz und Nieren geprüft.

Geliefert wird der Osborne O6T-386 mit MS-DOS 3.2, GW-Basic 3.22 und dem Ontrack-Disk-Manager 3.1. Wie bei allen 386er-Rechnern sind die Abmessungen des Osborne alles andere als mikro (B × H × T: 54 × 16,5 × 42 cm³).

Im Gegensatz zu vielen Rechnern der PC- und der AT-Klasse ist er jedoch mechanisch wesentlich robuster aufgebaut.

Ohne zunächst bei verschiedenen Programmen Zeitmessungen vorzunehmen, hinterläßt er sofort den Eindruck, daß er schneller als ein AT arbeitet.

Das Tastatur-Layout ist weitgehend identisch mit der MF-II-Tastatur von IBM. Ein abgesetzter Cursorsteuerblock und 12 Funktionstasten gehören heutzutage zum Standard. An der Tastatur ist, außer daß sie beim Tippen etwas laut klappert, nichts auszusetzen.

Ingesamt kann die Systemeinheit neben zwei Festplatten drei Einbaugeräte in halber Bauhöhe aufnehmen. Unter dem 1,2-MByte-Laufwerk paßt noch gut ein zweites Laufwerk. Leider ist die Frontblende etwas unglücklich gestaltet, so daß der dritte Einbauplatz von außen unzugänglich ist. Osborne hat das Problem erkannt und bietet für Anwender, die drei Einbaugeräte benötigen – z. B. für zwei Disketten-Laufwerke und einen Streamer – einen anderen Gehäusedeckel an. Man sollte sich daher vor dem Kauf genau entscheiden, ob man sich lieber den formschöneren oder den praktischeren Gehäusedeckel zulegt.

Wie bereits bei Rechnern der AT-Klasse üblich, ist auf der Frontblende ein Schlüsselschalter zum Sperren der Tastatur untergebracht. Darunter liegen der Reset-Taster, die Betriebs- und die Festplattenzugriffsanzeige.



Bild 1. Der Osborne O6T-386 ist ein preiswertes 80386-System

Zahlreiche Steckplätze

Mit zwei PC-Steckplätzen, vier AT-Steckplätzen und zwei 32-Bit-Steckplätzen liegt der Osborne im Durchschnitt der 386er-Rechner. Einen PC-Steckplatz nimmt die Hercules-kompatible Grafik-Karte in Anspruch, einen AT-Steckplatz der Festplatten-/Floppy-Controller und einen 32-Bit-Steckplatz die 2-MByte-Speichererweiterungs-Karte. Damit sind insgesamt noch fünf freie Steckplätze vorhanden. Kritiker werden jetzt bemerken, daß ja nur noch ein PC-Steckplatz vorhanden sei. Doch abgesehen davon, daß man die verbleibenden AT-Steckplätze auch mit PC-Karten bestücken kann, ist es ein glatter Anachronismus in einem 32-Bit-Rechner PC-Karten einzustekken. Leider sind fast alle derzeit am Markt erhältlichen Video-Karten als PC-Steckkarte ausgeführt. AT-Steckkarten für die Bildschirmausgabe sind noch rar gesät. Als Ergebnis erhält man einen Rechner, der eine ganz gute Rechenkapazität hat, der Flaschenhals liegt jedoch bei der Bildschirmausgabe. So hat ein Programmlauf von ATPERF – einem Testprogramm der amerikanischen Fachzeitschrift PC-Tech-Journal [1] ergeben, daß der O6T-386 durchschnittlich 17 Wartezyklen beim Zugriff auf die Hercules-Karte ausführen muß! Bei einer älteren CGA-Karte, die wir einem 8088-PC entnahmen, mußte die CPU sogar 25 Wartezyklen durchführen.

Dies ist jedoch kein Manko des O6T-386 – hier befindet er sich in bester Gesellschaft mit vielen anderen 80386-Rechnern.

Bei aller Kritik darf man jedoch nicht vergessen, daß die absolute Geschwindigkeit der Bildschirmausgabe dennoch höher als bei einem 8-MHz-AT ist. Selbstverständlich kann der O6T-386 mit einer leistungsfähigeren Grafikkarte bestückt werden.

Einen Überblick über die Ausstattung des O6T-386 gibt die *Tabelle*. Auf bis zu maximal 16 MByte RAM darf man das System erweitern. In dem Testgerät war eine 2-MByte-Speichererweiterungs-Karte in einem 32-Bit-Steckplatz untergebracht. Eine 8-MByte-Speichererweiterungs-Karte mit den 1-MBit-Speicher-Chips soll in Zukunft angeboten werden.

Die Grundplatine ist mit 512 KByte RAM bestückt. Damit der Prozessor auf den Speicher mit 32 Bit zugreifen kann, hat der Hersteller insgesamt 16 Speicherbausteine mit der Organisation 64 KBit × 4 verwendet, wobei je 8 Bausteine einen Speicherblock bilden. Sowohl die Speicherbausteine auf der Haupt- als auch die auf der Speichererweiterungskarte haben eine Zugriffszeit von 120 ns. Diese wären für den 80386 jedoch viel zu langsam, wenn Osborne nicht eine spezielle Speichertechnik einsetzte, die man in Fachkreisen verschränkte Speicher oder Interleaved Memory nennt. Der Speicher wird dabei so in zwei Blöcke unterteilt, daß aufeinanderfolgende 32-Bit-Adressen immer im jeweilig anderen Block liegen. Im Idealfall greift der Prozessor niemals nacheinander auf die gleiche

Speicherbank zu. Damit gelingt es, eine unangenehme Eigenschaft der dynamischen RAMs zu „verstecken". Nach jedem Zugriff müssen dynamische Speicherbausteine eine kurze Ruhepause einlegen, während der das System nicht auf sie zugreifen darf. Diese Erholzeit, in Datenblättern als Precharge Time bezeichnet, begrenzt die Speicherzugriffsgeschwindigkeit. In einem System mit verschränktem Speicher kann der Hersteller langsamere und damit auch billigere Speicher als in einem System mit der Standard-Speicherarchitektur einsetzen.

Mit einem weiteren Trick hat Osborne nochmals „Zeit geschunden". Der 80386-Prozessor kann Speicheradressen einen Zyklus früher als benötigt auf den Adreßbus legen und damit sorgen, daß die Zeit zur Decodierung der entsprechenden Speicher-Chips nicht voll in die Zugriffszeit des Speichers eingeht.

Die Praxis zeigt jedoch, daß trotz der raffinierten Speicherarchitektur, der Prozessor relativ häufig durch Wartezyklen ausgebremst wird. Das bereits erwähnte Programm ATPERF belegt, daß bei RAM-Zugriffen durchschnittlich zwei bis drei Wartezyklen durchgeführt werden. Dies hat mehrere Ursachen, die jedoch nicht im Systemkonzept des Osborne OT-386 zu suchen sind, sondern in der Codierung realer Anwenderprogramme. Da der 80386 nicht gerade mit vielen Allzweck-Registern gesegnet ist, muß er häufig die Operanden eines Befehls aus dem Speicher holen. Die Operanden-Adressen liegen mit einer relativ hohen Wahrscheinlichkeit auf der derzeit aktiven Speicherbank – d. h. die Zugriffszeit der Speicherbausteine macht sich voll bemerkbar und der Prozessor muß Wartezyklen durchführen. Arbeitet der 80386 mit seinen internen Registern, so kann er während er einen Befehl ausführt, bereits weitere Befehle in die interne Befehlswarteschlange mit der vollen 32-Bit-Datenbusbreite hineinschaufeln. So lange im Programm kein Sprungbefehl vorkommt, wird damit eine hohe Verarbeitungsgeschwindigkeit bei dennoch preiswerten Speicherbausteinen erreicht.

Will man eine noch höhere Verarbeitungsgeschwindigkeit, so muß man sich einen 386er-Rechner mit Cache-Speicher zulegen. Ein Cache-Speicher ist ein sehr schneller statischer RAM auf den der Prozessor direkt zugreift. Ein Cache-Controller sorgt dafür, daß im Cache der Code bzw. die Daten stehen, auf die der Prozessor am häufigsten zugreift. Für einen solchen Rechner muß man jedoch ein paar tausend Mark mehr hinlegen. Osborne hat mit dem vorgestellten Konzept einen Kompromiß zwischen hoher Verarbeitungsgeschwindigkeit und preiswerter Speicherarchitektur erreicht.

Tabelle: Die technischen Daten

Typ	O6T-386
Hersteller	Osborne
Geräteabmessungen	54 × 42 × 16,5
Speicher	
– Standard-RAM-Kapazität	512 KByte
– maximale Speicherkapazität	16 MByte
– Besonderheiten	2-MByte-Speichererweiterungskarte für 32-Bit-Steckplatz
Disketten-Laufwerk	
– 360 KByte, 5¼ Zoll	anschließbar
– 1,2 MByte, 5¼ Zoll	1
– 720 KByte, 3½ Zoll	anschließbar
– 1,44 MByte, 3½ Zoll	anschließbar
Festplatten-Laufwerk	ST 251
– Kapazität	40 MByte
– Zugriffszeit	40 ms
Erweiterungssteckplätze	
– PC-Steckplätze	2
– AT-Steckplätze	4
– 32-Bit-Steckplätze	2
– Anzahl der freien Steckplätze	5 (1 × 32 000 Bit, 3 × AT, 1 × PC)
Echtzeituhr	ja
Bildschirm-Schnittstelle	Hercules-kompatibel
Monochrom-Monitor	14-Zoll-Monitor (wird von Systemeinheit mit Netzspannung versorgt)
Parallel-Port	auf Hercules-Karte
Reset-Schalter	ja
Netzteil-Leistung	200 W
Tastatur	MF-II-kompatibel
CPU	80386
Taktfrequenz	16 MHz
Coprozessor	Fassung für 80387
Batterie	4 Mignonzellen
BIOS	Award 386 BIOS V2.09
Betriebssystem	MS-DOS 3.2
Sonstige Software	Diskmanager Ontrack GW-Basic 3.22

Testergebnisse

Programme wie Havard Graphics, Wordstar, Euroscript und Compiler (Turbo-Pascal, Turbo-C) liefen auf dem O6T-386 wesentlich flotter als auf Rechnern der AT-Klasse. Die Ausführungszeiten der mc-Testprogramme PRIM1, PRIM2 [2] sind in *Bild 2* dargestellt. In *Bild 3* stehen die Laufzeiten der Programme „Sieb des Eratostenes" (Sieve), „Fibonacci" [3] und die Ausführungszeit eines speziellen Befehls des Textverarbeitungsprogramms Euroscript. Die Programme „Sieb des Eratostenes" und „Fibonacci" wurden mit dem Turbo-C-Compiler compiliert. Alle verwendeten Testprogramme enthalten keinen 32-Bit-Code. Dies ist hier kein Fehler, da die überwältigende Mehrzahl der derzeit erhältlichen Programme ebenfalls keinen 32-Bit-Code enthält.

Beim Euroscript-Test wurde bei einem typischen mc-Beitrag (Dateigröße: 10785 Bytes) mit dem globalen Ersetzungsbefehl „GK" der Buchstaben „e" einfach durch einen anderen Buchstaben ersetzt und die benötigte Zeit gemessen.

Die Geschwindigkeitsunterschiede zwischen einem 10-MHz-AT (hier: mc-modular-AT ohne Wartezyklen) und dem Osborne sind etwas kleiner als erwartet. Hier macht sich das Fehlen eines Cache-Speichers bemerkbar. Dennoch arbeitet das System spürbar schneller als z. B. der mc-modular-AT und ist einem Standard-AT (8 MHz) haushoch überlegen.

Zubehör en masse

Auf der Grundplatine ist der Arbeitsspeicher nur 512 KByte groß. Will man das System auf 640 KByte erweitern, muß man sich eine entsprechend langsame AT-Speichererweiterungskarte zulegen oder auf die von Osborne angebotene 2-MByte-Speichererweiterungskarte (mit 32-Bit-Bus) zurückgreifen. Als Zubehör gibt es einen PAL-Baustein, der die Karte so decodiert, daß 128 KByte von der 2-MByte-Karte „abgezwackt" und dem DOS-Speicherbereich zugeordnet werden. „Lebensnotwendig" wird die 2-MByte-Karte allerdings erst, wenn man ein speicherfressendes Multitasking-Betriebssystem wie z. B. OS/2 einsetzen

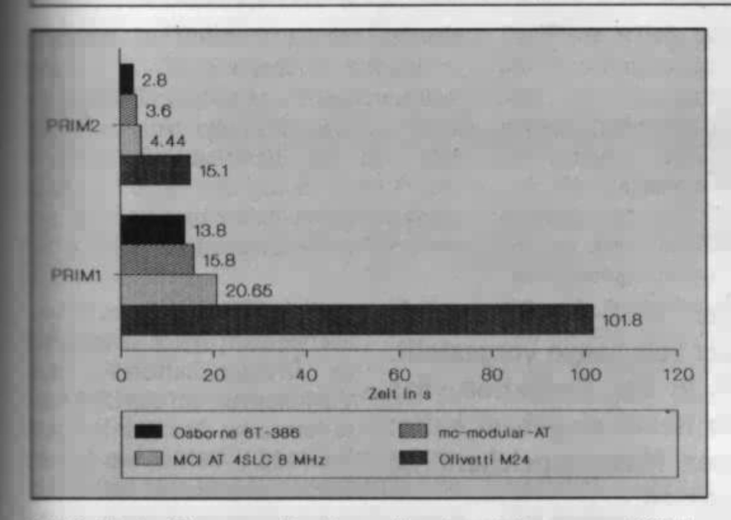


Bild 2. Die Ausführungszeiten der mc-Benchmark-Programme PRIM1 und PRIM2

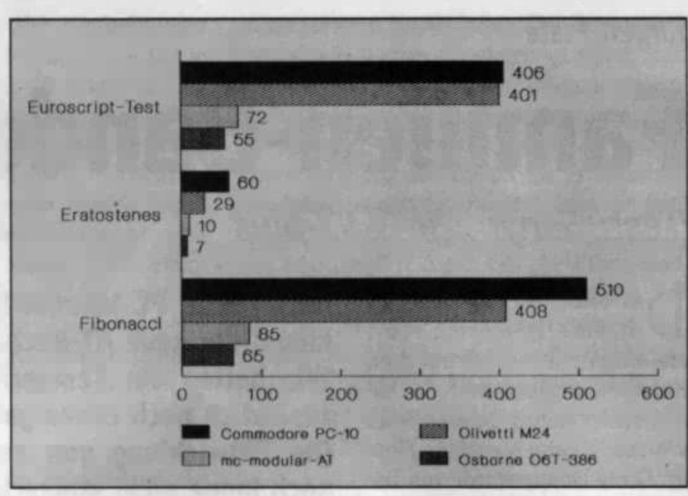


Bild 3. Auch im Euroscript-, Eratostenes- und Fibonacci-Test glänzt das System mit Schnelligkeit

will. Negativ zu vermerken ist, daß Osborne dem System keinen Expanded Memory Manager beilegt, mit dem man auch unter MS-DOS einen größeren Speicher ansprechen kann.

Wem die standardmäßig mitgelieferte Festplatte ST-251 (durchschnittliche Zugriffszeit 40 ms) zu langsam ist, bekommt für einen Aufpreis von 100 DM eine Festplatte mit einer mittleren Zugriffszeit von 26 ms. Wer schon einen 386er-Rechner braucht, sollte auf dieses Angebot zurückkommen. Positiv zu werten ist, daß man am Festplatten-/Floppy-Controller auch 3½-Zoll-Diskettenlaufwerke (720 KByte) anschließen kann. Damit kann der Anwender ohne Probleme Disketten von Laptops lesen. Spätestens zum Zeitpunkt der Veröffentlichung dieses Beitrags will Osborne auch ein 1,44-MByte-Laufwerk (PS/2-kompatibel) anbieten.

Weitere Grafikkarten sind für den O6T-386 ebenfalls erhältlich.

Fazit

Der Osborne 6T-386 ist für alle zu empfehlen, denen der Durchsatz eines Computers der AT-Klasse nicht ausreicht und die ein preiswertes 80386-System suchen. Für einen Preis von deutlich unter 8000 DM erhält man ein System, das sich als Grundgerät für die Bereiche Desktop Publishing, Software-Entwicklung, Grafikerstellung und als File-Server eignet. Auch für Programme, die eine grafische Benutzeroberfläche haben – wie z. B. Excel von Microsoft – ist der Osborne O6T-386 empfehlenswert. Er ist durchaus eine Alternative zu einem schnellen AT.

Literatur

[1] *Forgeron, Pierce, Armbrust:* Updating the Evaluation Suite. PC-Tech-Journal. März 1987, Seite 71.

[2] *Strauß, Dieter:* Primzahlen-Sucher. mc 1987, Ausgabe 10, Seite 74.

[3] High-Tech Horsepower. Byte Juli 1987, Seite 101.

Festplatten-Toolbox für Turbo-Pascal

MS-DOS ist recht spröde, wenn es um die Bearbeitung von Directories, der FAT (File Allocation Table) oder das direkte Schreiben von Sektoren geht und mancher Turbo-Programmierer ist über Prozeduren zum Lesen des Inhaltsverzeichnisses nicht hinausgekommen. Aus Österreich kommt von Glaser, Wien, eine Toolbox mit einer ganzen Palette von Routinen zur Dateiorganisation auf Disketten und Festplatten. Die Programme und Routinen lassen kaum Wünsche offen. Aus den meisten Routinen wurde ein Demo-Programm zusammengestellt, das auch als COM-Datei zum sofortigen Ausprobieren vorliegt. Es führt die Leistung der Diskbox vor: Reorganisation der Blöcke und des Directory zur Beschleunigung des Zugriffs. Nach der Reorganisation liegen alle Blöcke einer Datei sequentiell hintereinander, was das Lesen oder Laden einer Datei enorm beschleunigt. Man kann sich aber auch nur ansehen, wie stark die Dateien auf der Platte verteilt sind – ich war beim Test ganz schön überrascht, wie wüst die Cluster der verschiedenen Dateien auf meiner Festplatte verstreut waren.

Weiterhin ist das Sortieren von Inhaltsverzeichnissen möglich, nach Name, Erweiterung (Extension), Datum oder Größe. Test auf Lesefehler und markieren defekter Sektoren, Zugriffszeitmessungen und physikalisches Löschen von Dateien werden in Beispielen gezeigt. Das Demo-Programm ist menügesteuert. Es ist bekannt, daß Programmierer nur ungern dicke Handbücher verfassen. Schließlich ist jedes (!) Programm ganz einfach zu bedienen – oder? Der Hersteller der Diskbox bildet da eine rühmliche Ausnahme, denn das Handbuch im Ringordner hat ungefähr 200 Seiten. Nach einer kurzen Einführung in den gesamten Problemkreis erfolgt eine tabellarische Vorstellung der Routinen. Danach werden die einzelnen Konstanten, Typen, Variablen, Prozeduren und Funktionen ausführlich und sehr übersichtlich erläutert. Alle Routinen und das Demo-Programm sind im Quelltext auf der Diskette enthalten. Die Routinen dürfen in eigenen Programmen verwendet werden. Man darf Programme, die mit der Diskbox erstellt wurden, als COM-Datei ohne Lizenzgebühren weitergeben – nicht jedoch das Quellprogramm! Der Anwender lernt bei der Diskbox eine Menge über die Dateiorganisation unter MS-DOS.

Bevor man anfängt auf der Festplatte herumzubasteln, sollte man den Hinweis im Handbuch beherzigen und eine Sicherungskopie des gesamten Festplatteninhalts anfertigen. Programmierfehler können beim direkten Plattenzugriff schnell zum Datenverlust führen. Auch kopiergeschützte Programme sollte man deinstallieren, bevor man komplexe Festplatten-Utilities programmiert. Diese Regeln gelten nicht nur für die Diskbox.

Die 210 DM für die Diskbox sind keine Geldverschwendung, und an dem Handbuch sollten sich andere Toolbox-Produzenten ein Beispiel nehmen. *Jürgen Plate*

Jürgen Plate

Familien-Bande

Test: Sanyo MBC-17-plus

Nach dem PC-kompatiblen MBC-16-plus wurde jüngst der neue AT-Rechner von Sanyo vorgestellt. Wir hatten ein Testgerät in der Redaktion, das zusätzlich noch etwas ganz Neues eingebaut hatte: Die Entwicklung von neuen Massenspeichern ist noch lange nicht abgeschlossen.

Das Neue am MBC-17-plus ist eine 10-MByte-Floppy, die anstelle der sonst üblichen Harddisk eingebaut war. Sie haben schon richtig gelesen – in Worten: Zehn-Megabyte-Floppy. Diese Floppy wurde von Data Technology Corporation zusammen mit Sanyo und dem Diskettenhersteller Verbatim entwickelt. Doch dazu später noch mehr – sehen wir uns zunächst den „normalen" 17-plus an.

Der MBC-17-plus

Wie schon beim kleinen Bruder, dem PC-kompatiblen 16-plus, ist das Gehäuse sehr kompakt und findet mit 17,3 cm Breite, 33,7 cm Höhe und 41 cm Tiefe auch auf dem kleineren Schreibtisch Platz (*Bild*). Man kann das Gehäuse wahlweise ganz normal horizontal oder aber hochkant aufstellen (entsprechende Gehäusefüße sind bereits vorhanden). Der Hauptschalter befindet sich auf der Frontseite des Computers, es gibt also kein Gefummel auf der Rückseite des Gerätes. Auch die Tastatur wird von vorn eingesteckt. An die Rückseite des Geräts muß man also nur zum Anschließen von Drukker- und Schnittstellenkabel. Bedingt durch die kompakte Bauform muß der Schlüsselschalter für das Blockieren der Tastatur ein kleineres Modell als üblich sein. Der Schlüssel dazu nimmt kaum noch Platz am Schlüsselbund in Anspruch.

Der 17-plus ist in verschiedenen Ausbaustufen zu bekommen. Allen Rechnern gemeinsam ist die Grundausstattung: der Prozessor 80286, der wahlweise mit 6 MHz oder 8 MHz getaktet wird, ein Arbeitsspeicher von 512 KByte, der auf 1 MByte ausgebaut werden kann. Da das DOS von sich aus nur 640 KByte verkraftet, lassen sich die restlichen 384 KByte als RAM-Disk oder EMS-Zusatzspeicher ansprechen. Die Taktfrequenz ist per Software umschaltbar, ebenso kann man zwischen 0 oder 1 Wait-State wählen. Das Gehäuse bietet Platz für drei Slim-Line-5¼-Zoll-Laufwerke. Man kann neben dem obligatorischen 1,2-MByte-Laufwerk und der Festplatte noch ein Standard-Laufwerk oder ein 3½-Zoll-Laufwerk einbauen. Weiterhin sind eine serielle Schnittstelle, eine parallele Druckerschnittstelle und die Grafikkarte für den Bildschirm eingebaut.



Preiswert und kompakt ist der Sanyo MBC-17-plus

Beim Öffnen des Gehäuses konnte ich feststellen, daß auch im 17-plus von der herkömmlichen Technik Abschied genommen wurde: Der Sanyo ist auf einer Bus-Platine modular aufgebaut. Der Prozessor mit seiner gesamten Peripherie befindet sich auf einer langen Steckkarte, die auf den Bus gesteckt wird (das gleiche Prinzip haben wir kürzlich auch beim mc-modular-AT vorgestellt). Man kann den Computer also später hochrüsten – zum Beispiel mit einer 80386-Karte. Ein Steckplatz für den Arithmetikprozessor 80287 ist auf der CPU-Karte vorgesehen. Der Bus besitzt sechs Steckplätze, von denen drei durch CPU-Karte, Floppy/Harddisk-Karte und Grafikkarte belegt sind. Für Erweiterungen bleiben also noch drei Plätze frei, was in den allermeisten Fällen ausreicht, da Druckerschnittstelle und V.24-Interface auf der CPU-Karte vorhanden sind. Das Netzteil kann 138 W liefern, was ausreichend dimensioniert ist.

Die Grafikkarte ist umschaltbar. Man kann sie wahlweise als IBM-Farbgrafikkarte (CGA) mit einer Maximalauflösung von 640 × 200 Punkten oder als Hercules-kompatible Monochromkarte mit 720 × 348 Punkten betreiben. Über die Bildschirmwiedergabe ist nicht viel zu sagen; unser Testgerät war mit einem Sanyo-Farbmonitor ausgestattet, der ein ruhiges und angenehmes Bild lieferte. Die DIN-Tastatur mit 102 Tasten besitzt das neue Layout. Es gibt 12 Funktionstasten, die in einer Reihe oberhalb der alphanumerischen Tastatur angeordnet sind. Die Cursortasten sind als eigener Block zwischen Alpha-Tastatur und Ziffernblock angeordnet. Mir fiel auch auf, daß der 17-plus nach dem Booten die NUM-LOCK-Taste automatisch setzt. Neben dem Ziffernblock ist eine zusätzliche Return-Taste zu finden. Die Funktionen Caps-Lock und Num-Lock werden durch Leuchtdioden in den entsprechenden Tasten angezeigt. Der Tastenanschlag ist weich und angenehm. Das Geräusch des obligatorischen Lüfters und der Laufwerke blieb unter dem Pegel vieler Konkurrenzprodukte.

Der 17-plus ist im Besitz des Funkschutzzeichens, des TÜV-GS-Siegels und einer ZZF-Nummer – dem Anschluß von Modems, Btx-Zusatzkarten oder FAX-Karten steht also nichts im Wege.

Software

Der MBC-17-plus wird mit der MS-DOS-Version 3.2 und GW-Basic ausgeliefert. Im Lieferumfang sind Treiber für verschiedene Laufwerktypen (360 KByte, 720 KByte, 1,2 MByte, 8"-SD, 8"-DD, Hard-disk, Streamer), für das Einrichten einer RAM-Disk und für Festplatten mit bis zu 96 MByte

enthalten. Zur Unterstützung der Festplatte gibt es ein Vor-Formatierungsprogramm, ein FDISK für Laufwerkpartitionierung über 32 MByte und ein Parkprogramm. Beim Test auf Kompatibilität mit den klassischen Programmen gab es keine Probleme. Peter Nortons Sysinfo-Programm zeigte bei 8 MHz Taktfrequenz einen Wert von 9,2.

Die 10-MByte-Floppy

Die war nun wirklich etwas Neues! Sie heißt „Take Ten" und besteht aus einem speziellen Laufwerk, das die Größe eines normalen Slim-Line-Floppylaufwerks besitzt. Ein spezieller Controller befindet sich auf einer kurzen Einsteckkarte. Der Einbau einer Take-Ten ist also ebenso einfach wie bei einer Harddisk. Auch die Platten haben das gewohnte 5¼-Zoll-Format. Die eigentliche Diskette befindet sich aber noch einmal in einem harten Gehäuse (also so ähnlich wie bei den 3½-Zoll-Disketten). Auch der Schieber zum Verschließen der Schreiböffnung fehlt nicht. Auf der Platte befinden sich zwischen den Datenspuren zusätzliche Synchronisationsspuren mit analoger Aufzeichnung. Die Platten werden vom Hersteller vorformatiert geliefert und kosten etwa 130 DM pro Stück. Der komplette Kit wird für ca. 2500 DM angeboten. Die technischen Daten:

Die Platte rotiert mit 600 Umdrehungen pro Minute und besitzt 602 Spuren – 301 je Seite. Jede Spur enthält 78 Sektoren. Die mittlere Zugriffszeit beträgt 65 ms. Sie liegt also irgendwo zwischen Floppy und Harddisk. Ein Kopierversuch unter Betriebsbedingungen (ca. 100 KByte in etwa 50 Dateien) mit dem Copy-Kommando ergab eine Übertragungszeit von knapp 160 Sekunden. Die maximale Transfer-Rate beträgt 180 bis 275 KByte/s. Sie sehen, daß beim Zugriff auf eine Platte viele Dinge eine Rolle spielen und die Transfer-Rate gar nicht so wichtig ist.

Besonders interessant ist die Take-Ten sicher in der Ausbildung – jeder Schüler oder Student kann seine „Festplatte" mitnehmen. Auch dort, wo es um den Datenschutz geht (Geschäftsdaten, Personaldaten, Forschungsunterlagen usw.), ist die 10-Mega-Floppy genau richtig eingesetzt.
Mit dem Einbaukit werden auch die Treiber- und Installations-Software sowie ein spezielles Kopierprogramm geliefert. Betreibt man die Take-Ten als Laufwerk C, kann man das DOS auch direkt von der 10-Mega-Floppy booten.

Und sonst?

Für den Tester sind sie nicht so wichtig, für den Anwender dagegen um so mehr – die mitgelieferten Handbücher. Der Sanyo wird mit einem dicken Ringbuch geliefert, das die komplette deutsche Dokumentation enthält. Die Übersetzung ist sehr ansprechend. Nach einer Einführung, die neben der Aufstellung des Geräts die grundsätzliche Inbetriebnahme erläutert, folgt die ausführliche Beschreibung der DOS-Kommandos und des GW-Basic. Für die Profis sind zusätzlich alle DIP-Schalter (auch die auf der Hauptplatine), die Steckerbelegung, der Einbau des 80287 und sogar die Interrupt-Vektoren und die BIOS-Aufrufe beschrieben. Die englischen Handbücher liegen ebenfalls bei.

Zum Schluß vielleicht noch ein paar Worte zu den Preisen: Der 17-plus kostet mit einem Laufwerk, ohne Harddisk, Grafikkarte und Monitor ca. 2300 DM, mit zwei Laufwerken ca. 2800 DM, mit 20-MByte-Festplatte und einem Laufwerk ca. 3500 DM und mit 60-MByte-Platte ca. 4800 DM. Für die Grafikkarte muß man ca. 300 DM bezahlen. Man kann den Computer aber auch mit der EGA-WONDER-Karte (600 DM) bekommen. Ein Multisync-Monitor (CRT 100) ist ab Dezember lieferbar.

Der MBC-17-plus ist ein ausgereiftes Gerät zu einem ordentlichen Preis, das man bei der Kaufentscheidung auf jeden Fall berücksichtigen sollte. □

Bildschirme höchster Güte

Selbst den höchsten Ansprüchen genügen die neuen ETAP-Monitore, die von der RFI-Elektronik GmbH aus Mönchengladbach über 400 Händler exklusiv in Deutschland vertrieben werden.
ETAP Information Technology ist ein bei uns noch wenig bekannter Spezialist für Licht und professionelle Beleuchtung aus dem belgischen Malle. 1948 wurde ETAP als elektro-mechanischer Reparaturbetrieb in Antwerpen gegründet und entwickelte sich seitdem zu einem Unternehmen mit drei selbständigen Bereichen. Außer Beleuchtungsanlagen und unsinkbaren Segelbooten entwickelt und produziert ETAP seit Jahren Textverarbeitungssysteme.
1986 führte ETAP den DIN-A4-Monitor ETAP-Neftis und 1987 den DIN-A3-Monitor Atris ein, die jetzt beide von RFI vertrieben werden. Beide Monitore zeichnen sich durch ein flimmerfreies (75-Hz-Bildwiederholfrequenz) monochromes Bild aus.

Der DIN-A4-Monitor ETAP-Neftis mit seiner 15-Zoll-Bildröhre stellt auf der Mattscheibe schwarze Buchstaben auf rein weißem Hintergrund dar. Die Auflösung des ETAP-Neftis von 720 × 728 oder 720 × 1456 Bildpunkten garantiert eine besonders feine Linierung.
Für das gleiche Einsatzgebiet gedacht ist der ETAP-Atris, der mit seiner 20-Zoll-Röhre DIN-A3-Format und eine Auflösung von 1440 × 728 Bildpunkten erreicht.
Für die Entspiegelung hat ETAP eine Spezialbeschichtung entwickelt, wobei auf die Innenseite der Bildschirme eine Metalloxid-Dampfschicht aufgetragen wird. Dadurch werden Reflexionsabsorbtionsgrade von 60 bis 90 Prozent erreicht.



Das Beste vom Besten sind diese Monitore von ETAP, die es jetzt auch in Deutschland gibt

Die Monitore werden mit einer PC-Steckkarte geliefert, die den Hercules- und den CGA-Mode emuliert und einen Zwischenspeicher von 128 KByte bereitstellt.

Das Modell ETAP-Atris gibt es für etwa 7500 DM; der ETAP-Neftis kommt auf 5500 DM. Treiber für zahlreiche Anwenderprogramme sind verfügbar.

Peter Wollschlaeger

Omikron-Compiler für den Atari ST

... für den anspruchsvollen Programmierer

Der Basic-Interpreter von Omikron schlägt andere Basic-Interpreter für den Atari ST in der Verarbeitungsgeschwindigkeit deutlich. Mit dem Basic-Compiler von Omikron werden unglaubliche Geschwindigkeiten erreicht. Nur Assembler-Programme sind jetzt noch schneller.

Bis vor kurzem habe ich behauptet, daß Programme, die für Basic-Interpreter geschrieben sind, sich schon aus Prinzip nicht vernünftig compilieren lassen können. Außerdem werden die compilierten Basic-Programme von einem Programm in jeder echten Compilersprache, wie zum Beispiel C oder Pascal, deutlich geschlagen.
Da ich diese Meinung schon mehrmals schriftlich geäußert habe, sogar vor dem kritischen Leserkreis der mc, und niemand widersprochen hat, schien alles klar, bis heute...
Heute traf bei mir eines dieser Plastik-Bücher ein, in denen neuerdings Softwareprodukte mogelverpackt werden.

Aufschrift:
Omikron-Compiler für den Atari ST.

Untertitel:
... für den anspruchsvollen Programmierer.

Inhalt:
eine Diskette und ein Handbüchlein von 12 Seiten.

Letzteres ist tatsächlich ausreichend, denn die Bedienung des Compilers beschränkt sich auf das Starten des Compiler-Programmes und die Auswahl der zu compilierenden Source-Datei mit der von GEM bekannten File-Select-Box.

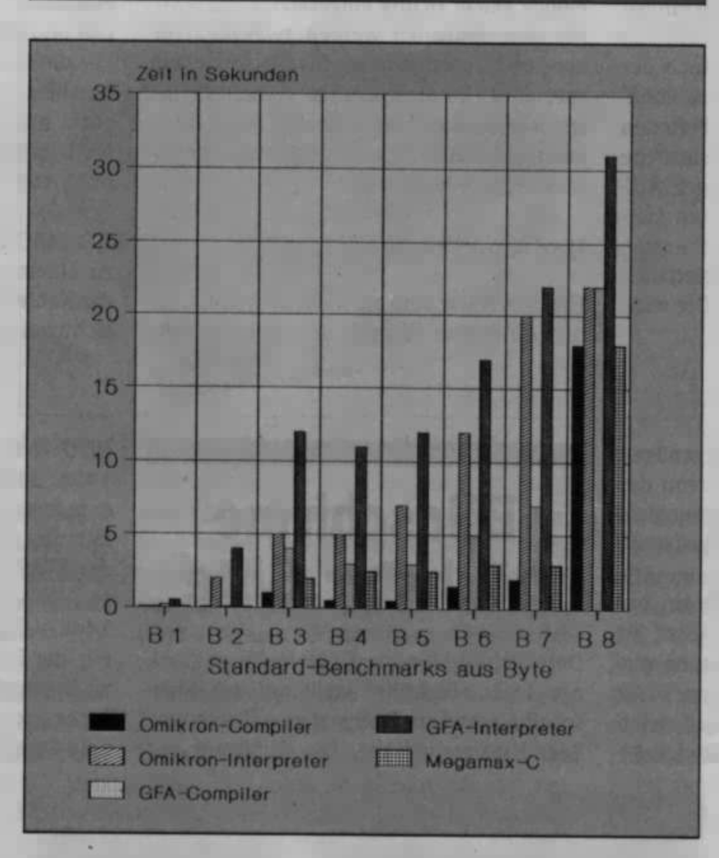


Bild. An acht Standard-Benchmarks zeigt der Omikron-Compiler, was in ihm steckt

Sehr schnelle Übersetzung

Im Gegensatz zu den meisten Basic-Compilern, die den Quelltext als reine ASCII-Datei erwarten, verlangt der Omikron-Compiler den Code, den der Interpreter erzeugt. Das erspart dem Anwender das Speichern mit der A-Option und bringt Zeitvorteile bei der Übersetzung in reine Maschinensprache. Der Parser hat es natürlich viel einfacher, wenn er statt langer Texte für die Basic-Schlüsselwörter nur noch mit Tokens umzugehen hat. Der Omikron-Interpreter arbeitet nicht mit den von den meisten Basic-Interpretern her bekannten Token sondern erzeugt aus den Basic-Befehlen einen FIT-Code, was soviel wie Fast Interpreting Technic heißt. Mathematische Ausdrücke werden z. B. sofort nach der Eingabe in umgekehrter polnischer Notation in den FIT-Code gewandelt. Der Compiler baut auf diesen schon optimierten Code auf. Auf jeden Fall werden so Programme mit einer Übersetzungs-Geschwindigkeit von 48 KByte Quelltext in der Minute compiliert. Die Übersetzungs-Leistung wie üblich in Zeilen anzugeben fällt schwer, denn typische Basic-Programme haben häufig mehrere Anweisungen pro Zeile. Je nach Programmierstil liegt deshalb die Übersetzungszeit bei 5000 bis 10 000 Zeilen pro Minute.
Wenn nun jemand fragt, warum zum Beispiel Microsoft nicht auf den Trick mit der Compilierung des komprimierten Codes gekommen ist, so ist die Antwort, daß Microsoft mehrere Interpretervarianten hat, die alle auf Token-Ebene nicht miteinander kompatibel sind.

Schneller als Megamax-C

Daß Basic-Programme, die der Omikron-Compiler übersetzt hat, schneller sind als solche, die der GFA-Compiler erzeugt, war zu erwarten; schließlich ist auch schon der Interpreter von Omikron schneller [1]. Was nun wieder erstaunt, ist aber die relative Steigerung. Praktisch werden die meisten compilierten Programme im Mittel um den Faktor 10 schneller als die Interpreter-Pendants. Wenn ich da an den Compiler BASCOM denke, der etwa einen Faktor von 2 bis 3 erreicht, verblüffen diese Zahlen doch.
Wer nun glaubt, daß hinter dieser Leistung ein bis auf das letzte Bit getrimmter Assem-

bler-Code steckt, täuscht sich (auch). Der Omikron-Compiler wurde in Omikron-Basic geschrieben und hat sich dann selbst übersetzt, was fast eine halbe Stunde gedauert haben soll, und damit den ersten Härtetest bestanden. Das beweist einmal mehr die enorme Leistungsfähigkeit dieses Interpreters, den man ja wohl inzwischen unwidersprochen als das beste verfügbare ST-Basic bezeichnen kann.

Meine ersten Benchmarks ließen denn auch arge Zweifel an der Präzision des ST-Timers aufkommen, denn es kam dabei heraus: Omikron ist meistens schneller als Megamax-C (und ich bin ein Fan von Megamax). Aber auch das Stoppen der Zeit von Hand brachte in etwa die gleichen Ergebnisse. Die Benchmarks sind weitgehend die üblichen aus Byte. Alle Integers wurden als Langworte deklariert. Bei Fließkommazahlen wurden die Formatunterschiede in Basic nicht bewertet. Benchmark 8 wurde in C in „double“ geschrieben („float“ wäre unfair). Die Zeiten sind handgestoppt. Teilweise wurden die Routinen in Schleifen gelegt, und die Ergebnisse dividiert, um noch messen zu können. Für diejenigen, die alles genau wissen wollen: Die Grafik wurde mit der Harvard Presentation Graphics auf einem PS/2 erstellt.

Wenn man „normal“ programmiert und über Integers oder die vier Grundrechenarten hinausgeht, ist Omikron immer schneller. Es lassen sich allerdings Benchmarks konstruieren, die das Megamax-C schneller erscheinen lassen. Das ist dann der Fall, wenn man sich auf „kurze“ Integers (16 Bit) beschränkt und zumindest die Laufvariablen als Register-Typ deklariert. Letztere Sache ist aber auch nur vorläufig, denn schon im Dezember 87 (also vor dem Erscheinen dieses Artikels) erhalten alle registrierten Omikron- Kunden (haben Sie Ihre Karte eingeschickt?) ein kostenloses Update, das auch Register-Variable bietet.

Mit CutLib kleiner Overhead

Das compilierte Programm erwartet im Normalfall die Datei BASLIB (die Runtime-Library) auf derselben Diskette vorzufinden. An sich ist das ganz praktisch, denn die eigentlichen Programmdateien bleiben damit fast genauso groß wie die ursprünglichen Basic-Programme. Wenn Sie jedoch ein Programm „stand alone“ brauchen, müssen Sie die Basic-Bibliothek dazu linken. Der Linker heißt nicht ganz zu Unrecht FULLIB, denn er bindet „ohne zu überlegen“ die volle Basic-Library ein, womit ein Basic-Programm um rund 40 KByte zunimmt. Bei Desk Accessories empfinde ich so etwas als ausgesprochen unfein und sah darin dann auch einen Grund, doch bei Megamax-C oder ST-Pascal zu bleiben. Offensichtlich muß man das bei Omikron geahnt haben, denn zum schon erwähnten Update gehört auch noch CutLib. Damit ist eine Routine gemeint, die aus BASLIB das herausschneidet, was das jeweilige Programm tatsächlich braucht.

Wenige Einschränkungen

Ich bin es ja schon gewohnt, daß bei Basic-Compilern das halbe Handbuch mit einer Liste von Inkompabilitäten zum Interpreter gefüllt ist, doch auch hier fällt Omikron positiv aus dem Rahmen. Die Einschränkungen der Reihe nach:

Bei Labels sind nur noch die ersten acht Zeichen signifikant. Stringvariable in FIELD-Anweisungen müssen global sein. Arrays müssen immer dimensioniert werden, auch wenn sie weniger als zehn Elemente haben. Das „wilde“ Verlassen von Programmstrukturen mit EXIT ist eingeschränkt. Für Funktionen muß der Typ des Return-Wertes angegeben werden. Bei Verwendung der „ON XXX“-Befehle (z. B. ON ERROR) muß ein Compiler-Switch (im Quelltext) gesetzt werden. Flag-Variable als Var-Parameter in Prozeduren erhalten keinen Wert von der Prozedur. Innerhalb von Prozeduren, die Arrays als Parameter übernehmen (typische GEM-Anwendung), dürfen keine neuen Arrays dimensioniert werden, auch Dateien dürfen an dieser Stelle nicht geöffnet oder geschlossen werden.

Als Vorteile gebenüber dem Interpreter sind zu nennen: Mit DIM können nun Felder auch verkleinert werden. Das von Omikron so genannte Multitasking (alle Befehle der Gruppe „ON event GOSUB“) wurde verbessert und kann nun auch im Betrieb aus- und wieder angeschaltet werden. Auch Interrupts via Ctrl-C sind nun möglich.

Fazit: Profi-Tool

Der Omikron-Basic-Interpreter ist an sich schon so schnell, daß für die meisten Anwendungen seine Geschwindigkeit mehr als ausreicht. Interessant ist der Compiler für die professionellen Anwender, für die auch gesparte Minuten noch „money“ sind, sowie für Leute, die „stand alone“-Programme vorziehen. Sicherlich ist das Produkt speziell für Entwickler und die in der Werbung angesprochenen anspruchsvollen Programmierer interessant, lassen sich doch hier auf eine sehr komfortable Weise Programme erstellen, die in Ihrer Leistung so manchen klassischen Hochsprachen-Compiler schlagen.

Literatur

[1] *Reiner Schönrock:* Omikron-Basic, mc 4/87.

Atari ST-Tips

Es ist mir fast ein wenig peinlich, daß ich schon so viele Listings in 68000-Assembler veröffentlicht habe, die an einer Stelle ineffektiven Code erzeugen. Da tröstet es auch wenig, daß ganze Heerscharen von ST-Programmierern den gleichen Fehler machen. Gemerkt habe ich das erst, als ich ein Handbuch von Hisoft übersetzt habe. Diese auf 68K-Assembler spezialisierten Profis hatten nämlich eine Idee, die zwar einfach ist, auf die man aber erst einmal kommen muß. Das Problem liegt beim Trap-Aufruf, einer der häufigsten Übungen bei jedem Programm für den Atari ST. Ein Trap-Aufruf läuft immer nach diesem Schema ab:

1. Parameter auf den Stack
2. Funktionsnummer auf den Stack
3. TRAP #x
4. Stack korrigieren

Das Problem liegt beim Punkt 4. Solange nicht mehr als 8 Byte auf den Stack gebracht wurden, ist die Sache einfach. Man schreibt:

ADDQ.L #n,SP

besser geht's nicht. Bei mehr als 8 Byte zu schreiben:

ADD.L #n,SP

ist nun allerdings nicht mehr optimal. Effektiver ist da schon: LEA n(SP),SP. Dieser Befehl kostet 4 Byte und dauert acht Clock-Zyklen. „ADD.L“ (korrekt ADDI.L) braucht zwar auch 4 Byte, dauert aber mit 16 Clock-Zyklen genau doppelt so lange. Bei der Gelegenheit:

LEA Adresse,An

ist zwar auch schon besser als:

MOVE.L #Adresse,An

denn das spart vier Clock-Zyklen, aber noch besser ist:

LEA Adresse(PC),An

Auch wenn ein Programm dadurch nicht lageunabhängig wird, so wird es doch kürzer und damit schneller von der Diskette geladen. Jede absolute Adressierung kostet nämlich einen Eintrag in der Relokatier-Tabelle, der bearbeitet werden muß.

Peter Wollschlaeger

Jürgen Plate

Festplatte sicherer gemacht

Disk Technician für MS-DOS-Computer

Eigentlich weiß der Anwender recht wenig über die Zuverlässigkeit seiner Festplatte, da MS-DOS bei einem Lesefehler bis zu 30mal den Lesezugriff wiederholt. Wenn eine Fehlermeldung auf dem Bildschirm erscheint, ist es aber meist zu spät. Das Programm Disk Technician verspricht hier Abhilfe.

Der Disk Technician kommt in recht ungewöhnlicher Aufmachung ins Haus. Diskette und Handbuch befinden sich in einer bunt bedruckten LP-Schallplattenhülle. Die Rückseite der Hülle gibt Aufschluß über die Notwendigkeit des Programms nach dem Motto: „Wenn Ihre Platte noch nicht abgestürzt ist, kann es sich nur um einen ausgesprochenen glücklichen Zufall handeln, denn jede Festplatte verliert Daten – es ist nur eine Frage der Zeit." In dieser etwas reißerischen Aussage steckt jedoch eine richtige Aussage, denn es gibt genügend mechanische und elektrische Faktoren, die bei Festplatten Lesefehler hervorrufen können. Wenn der Fehler im Boot-Sektor, in der FAT oder im Wurzel-Directory (Root-Directory) vorliegt, hat der Normalbenutzer kaum noch Chancen und auch der Profi kann mit den Norton-Utilities nicht immer alles retten. Das Handbuch, das derzeit leider nur in Englisch vorliegt, ist für Anwender, die keine Handbücher lesen, denn es besteht aus nur 16 Seiten. Und auch die muß man nicht lesen – es genügt die „60-Sekunden-Anleitung". Da Programm und Installation menügesteuert ablaufen und man fast nichts einzugeben braucht, reicht das Handbuch für die Bedienung aus. Ich hätte natürlich gern mehr über die Arbeitsweise des Programms erfahren.

Fehlerhafte Sektoren werden automatisch formatiert

Ein paar Informationen ließen sich trotzdem zusammentragen. Das Programm prüft jeden Sektor mit eigenen Leseroutinen. Sind die Daten nicht mehr ganz sauber zu lesen, wird die Information der entsprechenden Spur in den Hauptspeicher ausgelagert und diese Spur neu formatiert. Die Formatierung erfolgt sowohl auf der niedrigen Ebene der sogenannten Vorformatierung als auch auf der Ebene des Format-Kommandos. Danach werden die Daten wieder restauriert. Eine komplette Formatierung der Platte im Fehlerfall ist somit nicht mehr notwendig. Weiterhin kann das Programm Daten auf andere, freie Spuren auslagern und die nicht mehr lesbaren Spuren als defekt markieren. Was von den Herstellern als „automatisches System mit künstlicher Intelligenz" bezeichnet wird, ist eine kleine Datenbank, die über reparierte und unsichere Sektoren der Platte Buch führt. Diese Sektoren beobachtet der Disk Technician, und bei einer weiteren Verschlechterung der Lesbarkeit lagert er die darauf befindlichen Daten auf andere Sektoren aus. Ein Defekt wird bereits erkannt, bevor die Daten gänzlich verlorengehen. Das Programm lernt bei jedem Durchlauf hinzu. Ob es ein Regelsystem gibt, das Entscheidungen über eine Reparatur oder Umschichtung entscheidet, konnte ich mangels Hintergrundinformationen nicht herausfinden. Für den Anwender sind diese sicher auch nicht notwendig.

Einfache Installation

Die Diskette ist mit einem Kopierschutz versehen. Die Installationsroutine kopiert das MS-DOS und den COMMAND.COM auf die Originaldiskette und erzeugt eine AUTOEXEC.BAT-Datei. Das Handbuch empfiehlt, den Disk Technician zum Starten ins Laufwerk A zu schieben und einen Warmstart (Ctrl-Alt-Del) durchzuführen. Da auf der Diskette auch die Datenbank des Programms liegt, ist das Programm sowieso an eine bestimmte Platte gebunden. Der Kopierschutz soll wohl nur verhindern, daß in einer Firma von der jungfräulichen Originaldiskette Kopien gezogen werden, um alle Rechner zu bedienen. Bei einem Preis von knapp 300 Mark ist die Anschaffung einer eigenen Programmversion für jeden Rechner auch nicht allzu teuer.

Der Hersteller verdonnert den Anwender zur täglichen Verwendung des Programms, wobei es drei Modi gibt: täglich, wöchentlich und monatlich. Der erste Lauf dient dem Kennenlernen der Platte, es ist die Monatsversion. Diese kann man wirklich nur einmal im Monat ertragen – bei meinem AT mit 32-MByte-Platte dauerte der

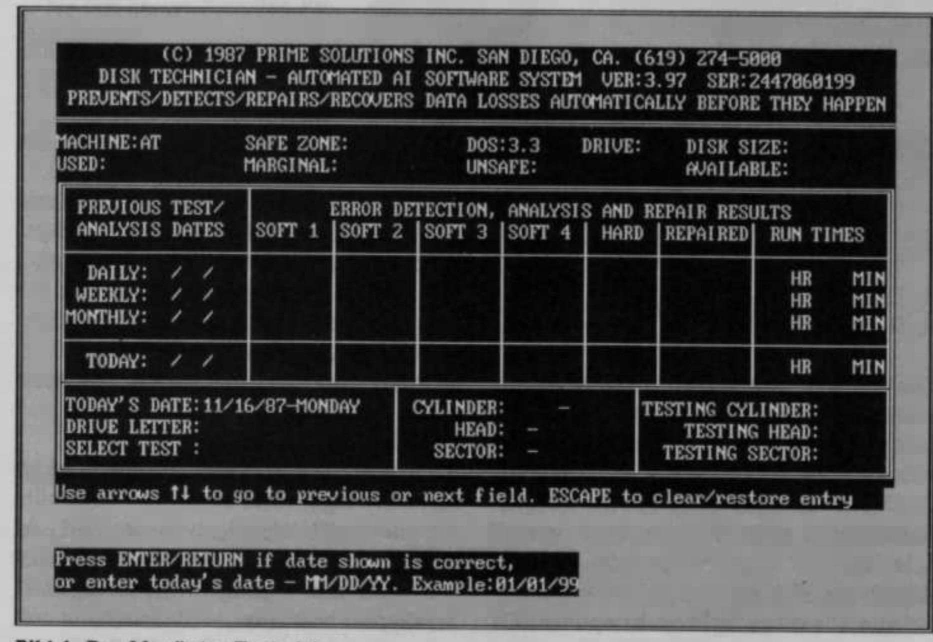


Bild 1. Das Menü des Technician

```
Cumulative preventions/detections/repairs/recoveries from 11/16/87 TO 11/16/87
        ¦  SOFT 1 ¦  SOFT 2 ¦  SOFT 3 ¦  SOFT 4 ¦  HARD ¦ REPAIRED
--------¦---------¦---------¦---------¦---------¦-------¦
FAULTS: ¦      25 ¦       0 ¦       0 ¦       0 ¦     0 ¦       25
 BYTES: ¦  12,800 ¦       0 ¦       0 ¦       0 ¦     0 ¦   12,800

    DATA SECTORS ¦      SYSTEM SECTORS ¦              TOTAL ERRORS
-----------------¦---------------------¦
FAULTS: ¦      0 ¦                   0 ¦                        25
 BYTES: ¦      0 ¦                   0 ¦                    12,800

Total potential problem bytes continually
being specially monitored by each Disk Technician run: 217,600

Disk Technician has finished. Remove Disk Technician diskette and store safely
Press ENTER/RETURN then reboot system for normal use
```

Bild 2. Mit diesem Report verabschiedet sich das Programm

allererste Lauf 3 Stunden und 46 Minuten. Das Startmenü ist in *Bild 1* zu sehen. Die tägliche Version des Ablaufs dauert nur ein paar Minuten, die wöchentliche einige 10 Minuten. Laut Handbuch ist das absolute Maximum 22 Stunden. Zum Schluß gibt das Programm auf dem Bildschirm oder dem Drucker einen Report über die geleistete Arbeit aus. Man kann so den Zustand der Arbeit verfolgen. Ich habe beim ersten Lauf den Bildschirm mit einem Auge im Blick behalten. Immer wenn ein Lesefehler auftritt, blinkt eine entsprechende Anzeige am unteren Bildschirm auf. Dabei fiel mir auf, daß bei meiner Platte die meisten Fehler bei Kopf Nummer 6 auftraten – wahrscheinlich ist dieser Kopf falsch justiert. Ich vermute, der Kopf „fliegt" bei meiner Platte etwas zu hoch. Schade, daß der Report auf solche Tatsachen nicht eingeht. Schön wäre auch eine einfache, grafische Darstellung der Defektstellen. Insgesamt sind auf meiner Platte 25 Fehler gefunden und repariert worden (*Bild 2*). Interessant ist die Meldung weiter unten im Report: 217 600 Bytes auf der Platte sind kritisch und werden besonders überwacht. Die Fehlerstellen lagen alle im belegten Bereich der Platte; sie wurden vom DOS also nicht erkannt.

Sicheres Parken der Festplatte

Safepark heißt ein Zusatzprogramm, das bei der Installation des Disk Technician auf der Festplatte installiert werden kann. Dazu wird beim Installieren eine Ladezone auf der Platte als defekt markiert. Das Programm Safepark wird über den Aufruf in der AUTOEXEC.BAT-Datei resident installiert und sorgt dafür, daß (wählbar) nach 1...15 Sekunden nach dem letzten Plattenzugriff der Kopf in diese Ladezone fährt. Wenn dann der Strom ausfällt, kann nicht mehr viel passieren. Mit dem Disk Technician wird die Wahrscheinlichkeit eines Datenverlustes geringer. Das entbindet den Benutzer aber nicht davon, regelmäßig ein Backup seiner Daten zu machen – was viele Anwender vergessen. Die sonst hohe Anwenderfreundlichkeit des Programms wird dadurch gesenkt, daß man immer die Programm-Diskette verwenden muß. Andererseits könnte ein auf der Festplatte befindliches Prüfprogramm selbst durch einen Plattenfehler unbrauchbar werden! In Deutschland wird das Programm von H&B in Tettnang vertrieben. □

Bildschirmattribute mit Turbo-Pascal einfach eingestellt

Vor einiger Zeit wurde in mc eine kleine Routine vorgestellt, die zeigt, wie man in Turbo-Pascal-Programmen die Bildschirmattribute mit dem ANSI-Treiber einstellt.

Ohne ANSI-Treiber geht es mit den Standardprozeduren Textcolor und Textbackground. Will man zum Beispiel auf inverse Darstellung umschalten, kann man sich entweder zwei Prozeduren schreiben, die jeweils Hintergrund- und Textfarbe umschalten (bzw. restaurieren) oder man baut die Aufrufe der beiden Standardprozeduren direkt ins Programm ein. Dies wäre sicher die schlechteste Lösung.

Mit der im *Bild* gezeigten Prozedur kann man alle Video-Modi über eine einheitliche Schnittstelle setzen. Die drei benötigten Farben für normalen Text, hervorgehobenen Text und Hintergrund lassen sich als Programmkonstante vereinbaren und somit leicht ändern. Dem Einbau weiterer Attribute steht nichts im Weg.

Jürgen Plate

```
const
  colback    : Integer = White;                 { BS-Farben }
  colschrift : Integer = Blue;
  colhigh    : Integer = LightBlue;

type
  BS_Mode = (normal,invers,high,highinv,blinking,invblink,hblink,hinblink);

Procedure SetVideo(Attribute : BS_Mode);
  { Bs-Mode einstellen  (der Wert 16 sorgt für Blinken) }
  begin
  case Attribute of
    normal   : begin Textcolor(colschrift); TextBackground(colback) end;
    invers   : begin Textcolor(colback); TextBackground(colschrift) end;
    high     : begin Textcolor(colhigh); TextBackground(colback) end;
    highinv  : begin Textcolor(colback); TextBackground(colhigh) end;
    blinking: begin Textcolor(colschrift+16); TextBackground(colback) end;
    invblink: begin Textcolor(colback+16); TextBackground(colschrift) end;
    hblink   : begin Textcolor(colhigh+16); TextBackground(colback) end;
    hinblink: begin Textcolor(colback+16); TextBackground(colhigh) end;
    end;
  end;
```

Eine Prozedur in Turbo-Pascal stellt die Bildschirmattribute ein

Prof. Dr. G. Schoffa

Professionelle Lisp-Interpreter für PCs

Zwei erschwingliche Produkte im Vergleich

Lisp ist nach Fortran die zweitälteste heute noch verwendete Computersprache. Während die meisten höheren Sprachen zu den prozeduralen Sprachen gehören, ist Lisp eine applikative und funktionale Computersprache. In Lisp sind daher die meisten KI-Programme geschrieben.

Man muß drei Bereiche innerhalb dieses Marktes unterscheiden. Der erste ist der industriell-professionelle Markt. Hier werden größere Unternehmen, militärische Stellen usw. mit der auf Lisp abgestimmten Hardware und mit der umfangreichen Lisp-Software von darauf spezialisierten Hardware- und Softwarehäusern bedient. Die an Lisp angepaßten Computer enthalten Lisp-Mikroprozessoren und haben auch sonst eine Struktur, die die Entwicklung und die Ausführung der Programme in Lisp optimal unterstützt. Dazu gehören Symbolics-3600, Xerox-Dandelion, Sun-3, TI-Explorer, aber auch der wissenschaftliche Rechner VAX750. Die Kosten für die Hardware und Software liegen zwischen 50 000 und 500 000 DM. Die Bedienung und die Programmierung dieser Geräte wird in speziellen Kursen erlernt. Die an Lisp-Maschinen ausgebildeten Programmierer verdienen in den USA jährlich 50 000 $ und mehr.

Auf der entgegengesetzten Seite liegen die sehr billigen Lisp-Interpreter, die als Public-Domain-Software 10 DM oder weniger kosten. Wir haben drei davon getestet und dabei festgestellt, daß sie nicht einmal für semiprofessionelle Zwecke geeignet sind. Die Laufzeit für das später angeführte Programm TAK ist bei XLISP 30mal länger als beim professionellen muLisp87. Sie sind sämtlich kein Common-Lisp, heute ein allgemein anerkannter Standard für Lisp. Mehr als 70% der für Common-Lisp vorgesehenen Befehle fehlen, desgleichen die Schnittstellen zum Betriebssystem und vieles mehr. Fast alle angeführten Benchmarks waren nicht ausführbar, weil die dafür nötigen Funktionen fehlten. Billige Lisp-Interpreter sind allenfalls geeignet, die Struktur von Lisp oberflächlich kennenzulernen. Lisp ist aber eine schwer erlernbare Sprache, und die Mühe lohnt sich nur, wenn man anspruchsvolle Anwendungen für symbolische und funktionelle Programmierung hat. Dazu braucht man ein gut entwickeltes professionelles Lisp. Bei einfacheren Aufgaben ist man mit Basic, Pascal oder Fortran besser bedient als mit einem schwachen Lisp.

Der dritte Markt, der hier bewertet wird, liegt zwischen diesen beiden Märkten. Bei niedrigen Preisen sind die PCs heute so leistungsfähig, daß sie mit guten Lisp-Interpretern auch im Bereich der professionellen Programmierung einzusetzen sind. Es werden daher immer mehr Lisp-Interpreter für PCs angeboten. In unserer Datenbank sind zwölf Lisp-Interpreter für PCs registriert. Wir haben davon zwei Common-Lisp-Versionen getestet:

muLisp87, Version 6.01, von SoftWarehouse für IBM-PC (XT, AT), Compaq, TI-Professional, Tandy-1000 und 2000, HP-150 und HP-110, Zenith-Z-100 u. a. MS-DOS-PCs, erhältlich über AI-Consult, Ettlingen.
Preis für den Interpreter: 684 DM.
Preis für den Interpreter und Compiler: 912 DM.

GCLisp 1.02 von Golden Hill Computers, für IBM-AT, begrenzt geeignet für IBM-PC, erhältlich bei Brainware, Berlin.
Preis für den Interpreter: 1425 DM.

Beide Lisp-Interpreter gehören zu den verbreitetsten in Europa.

Wie weit ist der Standard für Common-Lisp erfüllt?

Die Normierungsvorschläge für Common-Lisp wurden von einer Kommission in den USA ausgearbeitet [1, 2]. Einige der Vorschläge, aber auch die zu große Zahl der Anweisungen (ca. 593), sind umstritten. Der größte Teil dieser Vorschläge wird aber heute von fast allen Softwarehäusern akzeptiert, obwohl auch sehr teure professionelle Lisp-Interpreter nicht alle Funktionen und Strukturen des Common-Lisp-Standards enthalten.

Wie weit die von uns getesteten beiden Lisp-Versionen für PCs den Standard des Common-Lisp erfüllen, ist in *Tabelle 1* zu sehen. Die Tabelle enthält nicht alle Gruppen der Funktionen.

Beide Lisp-Interpreter enthalten nur etwas mehr als die Hälfte der Funktionen des Lisp-Standards. Daraus entstehen jedoch kaum Nachteile, weil beide die wichtigsten Funktionen enthalten. Die restlichen sind für die Programmierung nicht unbedingt nötig bzw. können individuell zugefügt werden. Bereits diese mehr als 300 implementierten Funktionen frustrieren den Anfänger. Der professionelle Programmierer sieht aber darin eine große Erleichterung bei der Programmierung.

Praktische Gesichtspunkte

Benutzt man eine teure Lisp-Maschine, so bietet sie natürlich eine recht komfortable Programmierumgebung. Bei den PCs kann man sie nicht erwarten, aber einiges an Schnelligkeit und Benutzerfreundlichkeit sollte man sich wünschen. Unsere Erfahrungen mit den beiden Lisp-Versionen sind in *Tabelle 2* enthalten.

Der Vergleich ist dahingehend zu ergänzen, daß jetzt auch für GCLisp ein Compiler erhältlich ist, allerdings durch den zusätzlichen Erwerb eines GCLisp-286-Developers (3408 DM), welcher nur mit dem IBM-AT und 3 MByte RAM verwendet werden kann. Desgleichen ist bei GCLisp die EGA-Grafik als Halo-Graphics für IBM-AT erhältlich (1368 DM). muLisp87 mit Compiler kostet 912 DM, die CGA-, Hercules- und EGA-Grafik sind darin enthalten.

Das Handbuch von GCLisp ist insoweit zu beanstanden, als nach der formalen Be-

schreibung der Anweisungen sehr wenige oder gar keine praktische Beispiele gegeben werden. Das rund 500 Seiten dicke Handbuch von muLisp87 enthält dagegen so viele Beispiele, daß man es sogar als Ersatz für ein Lehrbuch verwenden kann. Zum GCLisp wird deshalb das Buch von *Winston* und *Horn* „Lisp" [3] und das Buch von *Steele* [1] mitgeliefert. Die Hilfe ist aber begrenzt, weil das Buch von Winston und Horn weniger ein Lehrgang für Lisp als vielmehr eine Sammlung zahlreicher Programmbeispiele ist. Das Buch von Steele hat noch weniger passende Beispiele: Es ist lediglich die Beschreibung und Erklärung des Common-Lisp-Standards.

Beiden Lisp-Interpretern sind auf Disketten Lisp-Lehrgänge beigefügt, die interaktiv am Bildschirm zu benutzen sind. Der Explorer-Lehrgang bei GCLisp ist besonders gut und umfangreich. Der Lehrgang bei muLisp ist zwar auch gut, ist aber kürzer und ersetzt kein Lehrbuch. Für muLisp gibt es aber ein spezielles Lehrbuch [2].

Der Editor des GCLisp ist wesentlich komfortabler als der von muLisp. Allerdings nur ganztägig arbeitende Programmierer können die vielen Möglichkeiten des Editors bei GCLisp lernen und nutzen. Sporadisch arbeitende Lisp-Programmierer ärgern sich über die Ladezeiten und auch darüber, daß der größte Teil des Speichers mit dem Interpreter und dem Editor belegt ist. Der bei 512 KByte RAM verbleibende Arbeitsspeicher von etwa 20 KByte reicht nur für kleine Programme, tiefer gehende Rekursionen sind neben dem Editor nicht zu testen. Lädt man vorher die Systemsoftware für die Grafik, so kann selbst bei 512 KByte der Editor nicht mehr geladen werden. Die langen, den Benutzer frustrierenden Ladezeiten für den Editor des GCLisp sind dadurch bedingt, daß hier der Editor nur als Lisp-Quellcode vorliegt, dagegen bei muLisp zwar auch als Lisp-Quelle geliefert wird, aber vom Benutzer in EDIT.SYS übersetzt werden kann.

Der Zugang zum DOS, ohne dabei den Lisp-Interpreter zu verlassen, ist bei GCLisp nur begrenzt möglich. Bei muLisp kann man nicht nur DOS-Anweisungen aufrufen (DIR usw.), sondern auch zwischendurch EXE- oder COM-Programme ausführen. Das ist heute unbedingt erforderlich, weil man dadurch auch die in anderen Sprachen geschriebenen Programme an Lisp anbinden kann.

GCLisp verfügt nur über einfache Rechengenauigkeit. Das hat zur Folge, daß man bereits die Fakultät von 8 nicht rechnen kann. muLisp hat dagegen einen unbegrenzten Zahlenbereich, sowohl für Integer- als auch für Real-Zahlen. Man kann

Tabelle 1: Zahl der Common-Lisp-Funktionen im Vergleich zum Standard

	Lisp-Standard	muLisp-87 6.01	GCLisp 1.02
Spezifikation der Datentypen	3	–	2
Prädikatfunktionen	31	17	24
Kontrollfunktionen	67	37	53
Makrofunktionen	4	4	4
Symbolfunktionen	13	4	12
Numerische Genauigkeit für			
Integer	beliebig	beliebig	16 Bit
Bruchzahlen	beliebig	beliebig	fehlen
Numerische Funktionen	100	59	45
Character-Funktionen	36	22	17
Sequenz-Funktionen	44	30	11
List-Funktionen	96	87	50
Hash-Funktionen	7	0	1
Array-Funktionen	31	8	10
String-Funktionen	25	19	9
Input-Funktionen	13	11	7
Output-Funktionen	18	9	8
File-Funktionen	11	9	8
Gesamtzahl der Funktionen	593	329	303

Tabelle 2: Ergebnisse mit einem IBM-AT-02 (6 MHz und Harddisk)

	muLisp87 6.01	GCLisp 1.0
Speicherbedarf für Interpreter und Editor	70 KByte	490 KByte
Komfort des Editors	+ +	+ + +
Ladezeit für den Interpreter von Harddisk	1 s	55 s
Ladezeit für den Editor von Harddisk	1 s	160 s
Objektorientierte Programmierung (FLAVORS)	ja	nein
Direkte Schnittstellen zu DOS	+ + +	+
Grafikfähigkeit und Programmunterstützung		
CGS	ja	ja
Hercules	ja	fehlt
EGA	ja	fehlt
Programmierung der Fenster	+ + +	+
Eignung für numerische Rechnungen	+ + +	+
Compiler	ja	fehlt
Unterstützung durch das Handbuch	+ + +	+
Unterstützung durch den Lehrgang	+ +	+ + +
Unterstützung durch Beispiele	+ +	+ + +
Kopierschutz	nein	ja

z. B. die Fakultät von 1000 oder den Sinus auf 300 Dezimalstellen genau berechnen. Durch die Möglichkeit, muLisp an muMath zu koppeln, kann man über muLisp sehr komplexe mathematische Aufgaben lösen, unter anderem Matrizenrechnungen, Differential- und Integralrechnung.

Bei GCLisp sind besonders lobenswert die zahlreichen und umfangreichen Beispiele, die im wesentlichen eine Erweiterung zu den Programmierbeispielen im Buch von Winston und Horn [3] sind.

Rechenzeiten

Die Beurteilung der Programmiersprachen in bezug auf die Ausführungszeiten ist ein vielseitiges Problem. Für Lisp gibt es dazu eine sehr umfangreiche Studie und eine Sammlung von Lisp-Programmen von *R. P. Gabriel* [4]. Die meisten in diesem Buch aufgelisteten Testprogramme haben wir mit muLisp und GCLisp ausgeführt und die Rechenzeiten durch die Zeitfunktionen von Lisp gemessen. Eine Diskette mit diesen Programmen und mit sonstigen Demos für muLisp87 ist bei AI-Consult, Ettlingen, erhältlich.

Die von uns gemessenen Rechenzeiten für muLisp87 und GCLisp sind aus *Tabelle 3* zu ersehen.

Aus der Tabelle ist zu entnehmen, daß muLisp in allen Tests zwei- bis dreimal schneller ist als GCLisp. Nach der Compilierung sind die Zeiten bei muLisp noch drei- bis viermal kürzer und nähern sich an die Zeiten der VAX750. Verwendet man bei GCLisp den Developer 286, der eine Compilierung erlaubt, so erhält man wesentlich kürzere Rechenzeiten. Mit dem Developer wird aber der Bereich der konventionellen PCs verlassen, schon wegen der Kosten und

Tabelle 3: Benchmark für muLisp und GCLisp am IBM-AT-02, 6 MHz

Name des Programms	Zeit/s[1])				
	muLisp87	GCLisp Interpreter	VAX750 CL	Xerox Dand.	Symbolics 3600
(TAK 18 12 6)	33,67	81,12	2,69	1,67	0,60
(STAK 18 12 6)	40,21	86,29	6,21	4,66	2,58
(CTAK 18 12 6)	47,51	106,55	13,86	63,20	7,65
(TAKL 18 12 6)	190,97	620,44	12,35	14,00	6,44
(RUN-BOYER)	138,19	Stacküberlauf			
(BROWSE 10 10 4)	44,33	126,27	195,11	174,00	30,60
(DESTRUCTIVE 600 50)	74,15	168,57	11,30	17,58	3,03
(INIT-TRAVERSE)	22,14	52,46	35,44	48,00	8,62
(RUN-TRAVERSE)	45,20	136,54	217,21	181,00	49,95
(DERIV 1000)	65,41	164,22	24,50	23,90	5,12
(DDERIV 100)	71,79	174,83	32,90	33,30	5,24
(DIV2-ITERATIV 300)	40,40	150,05	14,32	23,80	1,85
(DIV2-RECURSIV 300)	51,50	168,40	9,07	24,80	2,89

[1]) Zeiten für andere Rechner siehe [4]

Tabelle 4: Zahl der Durchgänge durch einzelne Lisp-Funktionen

	Call	1 –	<	Bindung	Throw	Catch	Total
(TAK 18 12 6)	63609	47706					111315
(STAK 18 12 6)	63609	47706	63609	47709			222633
(CTAK 18 12 6)	63609	47706	63609		47707	47707	270338

des nötigen Ausbaus des AT auf 3 MByte. Der Lisp-Rechner Symbolix 3600 ist in bezug auf die Geschwindigkeit die absolute Spitze, was man aber bei dieser teureren Lisp-Maschine erwarten sollte. Die Zeiten für VAX750, Xerox und Symbolics sind aus [4] entnommen. Der Vergleich mit typischen Lisp-Maschinen hinkt zum Teil deshalb, weil diese oft die Programme sofort in übersetzter Form liefern und ausführen.

Für ein Lisp-Benchmark-Programm sind mehrere Testprogramme notwendig, weil verschiedene Funktionen zu prüfen sind: Listenmanipulationen, arithmetische Operationen, Binden, Iteration, Rekursion, Speicherbereinigung (*Garbage Collection*) usw. In *Tabelle 4* sind für die ersten Programme der Tabelle 3 die Lisp-Funktionen und die Zahl ihrer Durchläufe aufgelistet. Man sieht, daß jeder nachfolgende Test durch weitere Funktionen erweitert wird.

Die Funktion TAK prüft also die Geschwindigkeit der Funktionsaufrufe und das Abziehen von 1, bei STAK kommen der Vergleichsoperator < und die Bindung der Variablen, bei CTAK statt der Bindung die Ausgabe über Throw und Catch hinzu. Bei den weiter unten in der Tabelle 3 stehenden Programmen nimmt die Zahl der Funktionen sehr stark zu. Um welche es sich handelt und wie oft sie wiederholt werden, kann bei Gabriel [4] nachgelesen werden.

Für die praktische Anwendung von Lisp sind besonders solche Benchmarks interessant, die KI-Programme simulieren. Das Programm BROWSE generiert eine AI-Datei und arbeitet sie mit „Pattern Maching" ab. Hier schneidet muLisp87 auf dem AT besser ab als teure Lisp-Maschinen. Auch wenn es um mathematische Programme geht, z. B. iterative und rekursive Division, sind die Zeiten von muLisp nur zwei- bis dreimal länger als bei VAX oder Xerox. nach der Compilierung sind die Zeiten gleich oder besser. Beim Durchlaufen eines Baumes im Programm TRAVERSE ist die Zeit schon beim muLisp-Interpreter wesentlich kürzer. Der Grund ist die besonders effiziente Adressierung der binären Strukturen. Weitere Gründe für sehr kurze Rechenzeiten bei muLisp sind noch folgende: In den meisten Lisp-Interpretern werden die Programme als Listen in ASCII abgespeichert und zur Ausführung auch so eingelesen. Bei muLisp werden die Programme in ein sogenanntes „Token-Type-Format" oder in den „Distilled Code" umgeschrieben, wodurch im Vergleich zur Abspeicherung in ASCII nur ein Drittel an Speicherplatz benötigt und die Rechenzeit um etwa 20% reduziert wird.

Man sieht also, daß muLisp mit einem AT durchaus mit professionellen Lisp-Übersetzern mithalten kann. GCLisp erreicht diese Qualität erst mit dem Developer-286.

Zusammenfassende Bewertung

Wer die Programmiersprache Lisp nur oberflächlich kennenlernen möchte, kommt mit billigen Public-Domain-Lisp-Interpretern aus. Dabei ist allerdings zu beachten, daß sie nicht den heute verbindlichen Standard von Common-Lisp erfüllen und auch sonst eine sehr schwache Rechenleistung und Umgebung bieten.

Für anspruchsvoll oder professionell arbeitende Lisp-Programmierer ist muLISP-87 mit dem dazugehörigen Übersetzer das richtige Werkzeug. Zu beachten ist noch, daß muLISP mit Übersetzer, Editor, Grafik, Debugger usw. nur ein Bruchteil von GCLisp kostet.

GCLisp ist selbst für einen IBM-AT zu groß und zu langsam. Es ist aber ein sehr guter Lisp-Interpreter, wenn man den AT auf 3 MByte ausbaut und noch einiges in den Developer-286 und zusätzliche Software investiert.

Literatur

[1] *Steele, G. L.:* Common LISP – The Language, Refernce Manual. Digital Press, Digital Equipment Corp., 1984. Ein 465 Seiten starkes Buch über Common-Lisp-Standard.

[2] *Schoffa, G.:* Die Programmiersprache Lisp. Franzis-Verlag, München, 1987. Eine praxisbezogene Anleitung zum Erlernen des Programmierens in muLisp.

[3] *Winston, P. H.; Horn, B.:* LISP. Addison-Wesley, 1984. Ein kurzes Lehrbuch für GCLisp und dazu zahlreiche Programme für diverse Anwendungen. Nach eventuellen kleinen Veränderungen laufen diese Programme auch mit muLisp und GCLisp.

[4] *Gabriel, R. P.:* Performance and Evaluation of Lisp Systems. The MIT Press, Cambridge, Mass., 1986. Ein Standardwerk für Lisp-Benchmarks.



mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

PLATINEN-LAYOUT

Leiterplatten-Entwurf auf PC/XT/AT-Systemen

BOARDSTAR ist ein Software-Paket für Entwurf und Reinzeichnung gedruckter Schaltungen und bietet Ihnen professionelle Leistungsmerkmale zu einem fairen Preis. Komfortable mausgesteuerte Benutzeroberfläche - Leiterbahnführung im 1/40" Raster - Bearbeitung doppelseitiger Platinen bis 516 qcm - Verwaltung von Macro-Bibliotheken - Reinzeichnung mit jedem handelsüblichen Plotter.

ROUTSTAR ist der Autorouter zum BOARDSTAR und erstellt selbsttätig Layouts aus vorgegebenen Schaltungsbeschreibungen: Layout-Beschreibung durch Grafik- oder Textfiles - Berücksichtigung vorgegebener Designkriterien - Problemlose Verarbeitung von BUS-Strukturen.

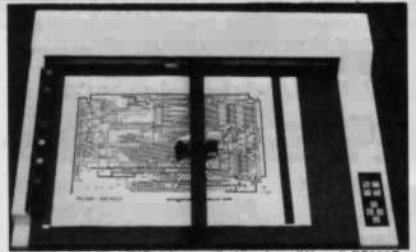
BOARDSTAR (Interaktiver Layout-Editor) .... 698,- DM
ROUTSTAR (Autorouter zum Layout-Editor) .... 698,- DM
BOARDSTAR-Paket (enthält BOARDSTAR + ROUTSTAR) .... 998,- DM
Lauffähiges Demo-Paket (wird angerechnet) .... 70,- DM
DIN-A3 Flachbett-Plotter PL-450 (8-Stift/HPGL/0,025mm) ..... 2450,- DM

Wir bieten Ihnen auch ein breites Spektrum von Service-Leistungen rund um das BOARDSTAR-Programm, vom Plot- u. Repro-Service bis zur Platinenfertigung nach angelieferten BOARDSTAR-Disketten. Fordern Sie Informationsmaterial oder besser gleich das Demopaket an.

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

TURBO-PLOTTER

Schnell, preiswert und präzise plotten mit PL-450



Format bis DIN-A3, HP-GL* kompatibel, serienmäßig mit RS232C und Centronics-Schnittstelle ausgerüstet, Pen-Speed maximal 565 mm/s, programmierbarer Stiftwechsel für 8 Farben (Pens), Aufnahme für HP-Standard Faser–/Keramik-/Tusche-Pens, Pen-Magazin mit Automatic-Verschluß, Auflösung 0,025 mm, verschied. Zeichensätze und und......

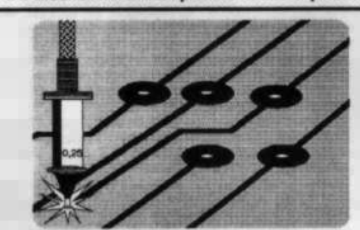
Unser Preis: .... 2450,00 DM

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

* HP-GL ist ein Warenzeichen der Hewlett-Packard GmbH

FOTOPLOTTER

Wir machen Ihren Penplotter zum Fotoplotter!



Fotoplot-Zusatzgerät LP2002, geeignet zum Anschluß an jeden Flachbett-Plotter. Gleichbleibende Schärfe und Strichbreite durch geschwindigkeitsabhängige Steuerung der Lichtintensität. Hervorragend geeignet zur Herstellung von Leiterplatten-Filmen mit CAD-Systemen.

Fotoplot-Zusatz LP2002 ....2223,00 DM

Fordern Sie Informationsmaterial an!

Datentechnik Dr.Gert Müller Diezstraße 2A
D-5300 Bonn 1
Tel. 0228/217297

Pocketmodem 1200 498,–
Nur 10 × 6 × 2,4 cm klein, für Portabels usw.

Modemkarte PC 1200 398,–
Kurze PC-Karte, V21/22, 300/1200 Bd vollduplex, RS 232 eingebaut (COM 1–4), Lautsprecher.

KS 1200, extern, sonst wie PC 1200 ..... **498,–**
KS 2400, 2400/1200 Bd vollduplex **698,–**
Multimodem, V21/22/23 .... **698.–**
Alle Modems Hayes-komp. und **ohne FTZ-Nr.**

RAMboard 2 MB 398,–
EMS-/Intel-komp., ohne RAM, mit Software
AT-RAMboard mit 16-Bit Slot, sonst w. o. **498,–**

AD/DA-Wandler 269,–
Auflösung 12 Bit, 16 AD-, 1 DA-Kanal

VIA 8255 179,–
I/O-Karte, 48 Leitungen, 16 LED, Timer/Counter

EGA Wonder Enhanced 598,–
Grafik, 800 × 560 Punkte max., mit Software

bsb
D. Schöddert – B. v. d. Brincken
Hohenzollernring 74
5000 Köln 1
Telefon **02 21-13 14 41**

Datentechnik



Der Eprommer für **IBM** und -kompatible **Apple** //e, //gs,][+, und **CPC 464/664/6128**

Programmiert alle gängigen EPROM- und EEPROM-Typen (z. B. 2716, 27C16, 2732, 2732A, 27C32, 2758, 2764, 2764A, 27C64, 27128, 27128A, 27C128, 27256, 27C256, 2508, 2516, 2532, 2564, X2804A, X2816A, X2864A ...) # Menügesteuerte Software auf Diskette/Kassette # 32 KByte frei für EPROM-Daten (Brennen des 27256 ohne Nachladen) # Kein Umschalten, Stecken oder Löten nötig # Programmierspannungen werden im Gerät erzeugt # Verbindung zum Rechner über Flachbandkabel # Rote und grüne LED zur Betriebsart-Anzeige # Komplett mit 28pol. Textool-Sockel # IBM- und CPC-Version mit Interface-Karte (durchgeführter Expansionsport bei CPC-Version)

Preise für IBM: Fertiggerät	**DM 399.50**	# Bausatz **DM 349.–**
für Apple: Fertiggerät	**DM 269.50**	# Bausatz **DM 219.–**
für CPC 464/664: Fertiggerät	**DM 289.50**	# Bausatz **DM 239.–**
für CPC 6128: Fertiggerät	**DM 319.50**	# Bausatz **DM 269.–**

CPC-Software auf 3"-Diskette statt Kassette: + DM 15.–

DOBBERTIN
INDUSTRIE-ELEKTRONIK GmbH
Brahmsstraße 9, 6835 Brühl, Tel. (0 62 02) 7 14 17

Industrie- u. Meßkarten für PC, XT, AT

– Copy-Board mit Laufwerkanschluß	229,–
– Bus-Extender-Karte f. PC, XT	98,–
– Colorgrafik- u. Herkules-Combi-Karte	298,–
– EGA-BAS-Adapter, wandelt RGB in SW-Video um	119,–
– IEEE-488-Karte zum Steuern für Meßgeräte	780,–
– RS 232 serielle Karte	89,–
– RS-422-Schnittstellenkarte	338,–
– Centronics-Karte für Druckeranschluß	89,–
– 35-MHz-AD/DA-Wandler, je 1 Kanal 8/10 Bit	548,–
– AD/DA, 12 Bit, 16 A/D- u. 1 D/A-Kanal	248,–
– AD-Wandlerkarte mit 8 Bit Auflösung, 1 Kanal	129,–
– Prototypen-Karte mit 24 TTL-I/O für XT, AT	198,–
– Relais-Karte mit 8255 u. 8 Relais m. Softw.	249,–
– Steppmotoren-Karte f. 2 Stepper m. Software	298,–
– Frequenzzähler von 300 Hz...1300 MHz f. PC/AT	298,–
– Digitalisierer m. 256 Graust. bis 1024 Pix.	991,–
– SSTV-Karte f. Bildübertragung per Funk m. AGC	328,–
– 48 I/Os mit 3×16-Bit-Zähler	248,–
– Optokoppler-Karte, 16 Eing./8 Ausgänge	368,–
– Eprommer 2716...27512 u. Textool-Sockel ext.	398,–
– PAL-Programmer für PC/XT/AT u. Software	798,–
– 640-KB-RAM-Karte, 0 KB best., für PC/XT	119,–
– 2-MB-RAM-Karte, 0 KB best., für PC/XT	329,–
– 3,5-MB-RAM-Karte, 0 KB best., für AT	339,–
– 4-MB-RAM-Karte, unbest.	389,–
– Modem-Karte f. Telefon, PC/XT, 300/1200 Bd. u. Software	339,–
– Game-I/O-Karte mit zwei Eingängen, kurze Karte	79,–
– Joystick für PC	36,–
– TVC-500, SW-Kamera mit Videoausgang, 220 V	448,–
– CCD-Industriekamera, 500 × 512 Pixel, ab 3 Lux	1799,–
– RS-232- auf RS-422-Wandler für lange Übertragungen	498,–
– Multiserielle Karte mit 4 ser. RS-232-Ports	229,–
– Serielle RS-232- u. Parallel-Printerport für XT u. AT	145,–
– 384-K-Multifunktions-Karte, 0 KB best.	189,–
– HDD/FDD-Controller, läuft bis zu 16 MHz	379,–
– Barcode-Pen mit vielen Normen u. Interfaces	850,–
– EPROM-Löschgerät für max. 6 EPROMs mit Timer	99,–

Bitte INFO anfordern. Händleranfragen erwünscht.
Versand erfolgt per NN. Auch Export möglich.

COMPUTER-RING
Heinrich Kolter Electronic, Postfach 11 27,
5042 Erftstadt-Lechenich, Steinstraße 22–24,
Telefon (0 22 35) 7 67 07, Telex 8 882 134



IDEAL

Der optimale Debugger für den Atari ST.
Kompatibel zum SID – um viele Funktionen erweitert:

- Volle Label-Verarbeitung
- Volles Screen-Editing
- Untrace mit Abbruchbedingungen u. Anzeige der zuletzt ausgeführten 64 Befehle
- Eingebauter 2-Paß-Assembler mit komfortablem Full-Screen-Editor

„Debugger der Superklasse"
(ST-Magazin Sept. 87, S. 74)

Unverbindliche Preisempfehlung **DM 99.–**



OMIKRON.Software, Erlachstraße 15,
D-7534 Birkenfeld 2, Telefon 0 70 82/53 86

MINIPREISE FÜR LAUFWERKE

PHILIPS X3132	2 × 40 Spur slim line	DM	249.--
PHILIPS X3134	2 × 80 Spur slim line	DM	270.--
	mit Umschaltung 40/80 Spur	DM	295.--
PHILIPS X3113	1 × 80 Spur 2/3 Bauhöhe	DM	120.--
	mit Umschaltung 40/80 Spur	DM	140.--
PHILIPS X3114	2 × 80 Spur 2/3 Bauhöhe	DM	249.--
	mit Umschaltung 40/80 Spur	DM	269.--
Floppygehäuse für slim line		DM	25.--
Tastaturen ohne Gehäuse		DM	25.50
Datenkabel für 2 Laufwerke		DM	32.--
Anschlußstecker für Stromversorgung		DM	2.95

Alle Preise zuzg. Versandkosten, Versand per NN oder Vorkasse

CHRISTEL VON DER LINDEN 4200 OBERHAUSEN
STERRADER STR. 189 TEL. 0208/663721 AB 14 UHR

---

mtr



mtr3, DER EPROM-HELFER

sekundenschnell — automatisch — handlich

- für EPROMs und EEPROMs bis 256Kbit (Option 1Mbit),
- Adapter selbst für wenig gebräuchliche Bauelemente,
- sekundenschnelle aber auch konventionelle Programmierung,
- Typwahl automatisch oder zumindest ohne Datenblatt-Hilfe,
- Emulatorfunktion
- V.24/RS232C-Schnittstelle, einfach konfigurierbar,
- Netz- und Akkubetrieb,

und das alles im kleinsten (189 x 138 x 48) autonomen EPROM-Programmierer für

DM 2257,20 **Sfr 1950,— exkl. Wust**

Meßtechnik Dr.-Ing. Ranfft
Dörpfeldstr. 15, D-5657 Haan 2, Tel. 0 21 04/6 28 27

Hedinger + Partner KG
Zentralstr. 3, CH-8610 Uster, Tel. 01/940 90 55

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.

Gesucht & Gefunden: BRAIN, die Literaturdatenbank

Ist es Ihnen nicht auch schon so ergangen: Sie erinnern sich nicht mehr, in welcher Zeitschrift ein bestimmter Artikel zu einem bestimmten Thema erschien? Hier hilft BRAIN:
BRAIN ist eine Literaturdatenbank, die speziell auf die Erfordernisse einer Literatur-Verwaltung abgestimmt ist und nicht unter den Mängeln, wie z. B. beschränkte Aufnahmekapazität pro Feld, „normaler" Datenbanksysteme leidet.
BRAIN bietet – unabhängig von der Eingabe – mannigfaltige Ausgabe-Optionen, seien es Autorennamen in Großbuchstaben oder die Vornamen nur als Initialen, die Jahreszahlen in Klammern oder den Titel nur bei Büchern unterstrichen.
BRAIN ist menü-geführt, eine Datensprache ist deshalb nicht erforderlich. Nach wenigen Minuten „Einarbeitung" kann's schon losgehen.
BRAIN bringt Ordnung in Ihre Literatur – zu einem wirklich akzeptablen Preis.

BRAIN – Literaturdatenbank für MS-DOS-Rechner (PCs und Kompatible), Software und ausführl. Handbuch **Preis 425.– DM**
Eine Demo-Diskette zu BRAIN ist für 20 DM erhältlich.



SHAMROCK SOFTWARE Vertrieb GmbH
Karlstraße 35, 8000 München 2
Telefon (089) 595468



C – COMPILER MANX

- Native-Compiler für MS–DOS, CP/M80, CP/M86, Apple, Amiga
- Cross-Compiler für 80186/286, Z80/8085, 6502 und 68000
- Compiler, Assembler, Linker, Hex–Converter, HLL-Debugger
- alle Tools zur Eprom-Erzeugung sind enthalten

Schema II Schaltplan – CAD

- superschneller Bildaufbau mit Zoom und Scrolling
- DESIGN–RULE–CHECK, Stücklisten- und Verbindungslisten
- Bauteile - Bibliotheken für Elektronik und E - Technik

PROTEL-PCB Platinen – Layout

SPITZENLEISTUNG ZUM SUPERPREIS !!!

- manueller Router mit 1/1000 Zoll Auflösung
- unterstützt Multilayer, SMDs, Gerberplots und Excellon
- automatische Netzlistenkontrolle mit Soll/Ist - Vergleich
- interaktiver Autorouter als Erweiterungspaket

Demodisketten für SCHEMA und PROTEL je DM 50,--

INGENIEURBÜRO DIPL.ING. MANFRED SUCHY
Gottlieb–Daimler–Str. 12 8037 Olching
Telefon 08142/12360 und 08142/28028



VIDEO-1000

Interface zum digitalisieren von Videobildern (TV, Kamera und Recorder) in 1/50 Sekunde (bei 2 oder 3 Graustufen/Farben).

VIDEO-1000 C, C-64, 384 × 288 Pixel, 2–4 Farben 295.– DM
VIDEO-1000 A, für APPLE II+, IIe, 384 × 288 Pixel . 295.– DM
VIDEO-1000 ST, für ATARI 260 ST...1040 ST, Monocrom und Farbe, 640 × 400 Pixel, 2 Graustufen; 640 × 200 Pixel, 4 Farben und 320 × 200 Pixel, 16 Farben .... 295.– DM
VIDEO-1000 I, für IBM XT/AT, unterstützt Colour-Grafic, Hercules- und EGA-Karten, 640 × 200 und 640 × 300 Pixel, 2–16 Farben .... 495.– DM

Info gratis. Demodisk nur gegen Einsendung von 5.– DM (APPLE, C-64, IBM) oder 10.– DM (ATARI) Schein oder Briefmarken. Der Versand der Digitizer erfolgt p. NN.

Ing.-Büro Manfred Fricke
Neue Str. 13, 1000 Berlin 37, Telefon 0 30/8 01 56 52

PC 80286

6/10 MHz, 1 MB RAM, 8 Slots, H.E.G.A.-Karte, 1,2 MB Floppy, 200-W-Netzt., d. Tastatur 102 T., Klappgeh. leg. DOS/BIOS .... **DM 2492.–**
Mit 20-MB-Festplatte .... DM 3050.–
40 MB, 28 ms .... DM 3650.–

Auch **80386** enorm günstig, ab **DM 5350.–**

14"-EGA-Monitor .... **DM 990.–**
EGA 800X600, Genod-Chip ... DM 399.–

2,5 MB RAM card, oK .... DM 195.–
30-MB-Festpl., set RLL, contr. Kabelsatz .... DM 720.–
AT .... DM 920.–

Epromer 4, textool-Software ... DM 420.–

Fölsche Computer
Wolfgang Fölsche · Hard- und Software
Untertorstraße 24 · 7022 L.-Echterdingen
Telefon 0711/793293

PCs und sämtliches Zubehör

mc–AllProg Plus

Universelles Programmiergerät aus mc 6–8/1987
mit erweiterter Software

+ Programmiert jetzt auch GALs.
  PROMs von TI und Zeropower-RAMs
+ Vorbereitet für Single–Chip–Prozessoren von INTEL
+ Unterstützt das INTEL–Hex–Format
+ Inhaltsverzeichnisanzeige

* Programmiert (E)EPROMs bis 27512, 27513 und 27011, PROMs von Signetics/Valvo und PALs von MMI, NS, TI und AMD
* Anschluß über RS 232 (V.24) Schnittstelle
* Bediensoftware für IBM–PC/XT/AT (unter PC/MS–DOS), kompatible und Z80–Systeme (unter CP/M) verfügbar
* Bildschirmorientierter (E)PROM–Editor
* Bildschirmorientierter Text–Editor zum PAL–Entwurf
* PAL–Assembler + PAL–Simulator + PAL–Disassembler
* Fuselisten als JEDEC–Dateien speicher–/lesbar

Bausatz (inkl. Handbuch und Software) ...... 997,– DM
Fertigplatine (getestet; Handb., Software) .. 1420,– DM

Versand per Nachnahme. Bei Bestellung unbedingt Diskettengröße, Diskettenformat und Betriebssystem angeben!

Holger Haase & Michael Menrad
Peripheriegeräte GbR
Postfach 2647 / 3300 Braunschweig
Telefon: (0531) 400310

Preis-Sensation

DIN-A3-Plotter
mit Papierbewegung
TSS 860
HP-GL-kompatibel
DM 3398,–
6 Farben
0,025 mm Auflösung
400 mm/s Zeichengeschwindigkeit
Centronics- und V.24-Interface
56 Zeichenbefehle
Lieferung per Nachnahme
TSS-Schmitz
In der Holl
5223 Bierenbachtal · Tel. 02293/2188
Inh. Brigitta Schmitz

mc 2/88

MC BASIC-EMUF

Neu: mc-GAL-Programmer aus mc 2/88, programmiert GAL 16V8 und 20V8 mit BASIC-EMUF oder IBM-PC

Bausatz GAL-Programmer o. TEXTOOL	DM 98,–
Fertigplatine GAL-Programmer o. TEXTOOL	DM 148,–
3M-TEXTOOL-Sockel-Modul, 24polig	DM 49,–
Fertiggerät im Gehäuse für IBM-PC	DM 348,–
Basic-EMUF	
BS1: Platine, GAL, deutsches Manual	DM 98,–
BS2: BS1 + IP80C52, XTAL1	DM 198,–
BS3: RTC58321, AKKU	DM 29,–
BS4: µPD7002, ZN458, XTAL2	DM 29,–
BS5: ULN2003A = 75468, 2× Relais	DM 29,–
BS7: BS1 + TL7705, MAX232, IP80C52, 74HCT08, 74HCT373, 71055, RAM8K	DM 238,–
BSLCDKEY: Bausatz komplett	DM 198,–
FPLCDKEY: Fertigplatine komplett	DM 248,–
Basic-EMUF-Fertigplatine, komplett mit 32 kB RAM	DM 638,–
Basic-EMUF-Fertigplatine, Lochrasterversion mit 8 kB RAM	DM 438,–
PASCAL-CROSS-COMPILER (MS-DOS)	DM 969,–
CROSS-ASSEMBLER (MS-DOS)	DM 869,–
CROSS-SIMULATOR (MS-DOS)	DM 948,–

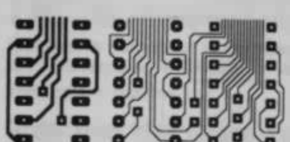
Intec Electronic GmbH
Rheingrafenstraße 37, 6501 Wörrstadt
Telefon 0 67 32/50 29

ATARI ST
NEC PINWRITER P5/P6/P7

PLATINEN-LAYOUT

MPK Printed Circuit Board Editor. Version 3.0 new!

Professionelles, voll mausgesteuertes Leiterplatten-CAD-Programm ohne Autorouter. Hardwarevoraussetzung: ATARI 1040 ST[F] mit einem NEC 24Nadel PINWRITER (auch MEGA ST möglich).
Platinen bis 203 · 240mm, beidseitig, 1/180 Inch Auflösung. Semifeinstleiter (zwei zw. benachb. IC-pins), 45°Grad.
Gesamtbildschirm-Fadenkreuzcursor mit einstellbarer Schrittweite. WYSIWYG und ein sehr schneller Bildaufbau erleichtern die interaktive Layouterstellung (Progr. läuft nur monochrom).
Jederzeit TOP VIEW oder BOTTOM VIEW (Spiegeln).
Ausdruck mit hoher Auflösung 1:1 für Prototypen oder 2:1 in Produktionsqualität (!garantiert! Sie ersparen sich den Photoplotservice!).
Ausfüllen der Lötaugenmittelpunkte beim Drucken möglich.
Getrennter Bohrplanausdruck (Lötstopmaske).
MPK PCB EDITOR mit deutsch. Handbuch DM 149,- zuzgl. Vers. per Nachnahme.
Originalausdruck:



MPK
Marek Petrik
Vogelsbergstr. 13
3550 Marburg 7
06421/47588

Händlerkonditionen auf Anfr.!

Preis-Sensation

mc 2/88

DIN-A3-Plotter
mit Papierbewegung nur DM
1990,-



HP-GL-kompatibel, 4 Farben
Serielle und parallele Schnittstelle
Lieferung per Nachnahme
TSS-Schmitz, Inh. Brigitta Schmitz
In der Holl, 5223 Bierenbachtal
Telefon (02293) 2188

mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.



ELINK-03 Datenfernübertragung ohne Probleme

- 300 BAUD vollduplex
- 1200 BAUD halbduplex
- 1200/75 BAUD vollduplex
- 75/1200 BAUD vollduplex
- ELINK-03 wählt vollautomatisch und nimmt Anrufe entgegen (MAILBOX)
- SMARTMODEM kompatibel: arbeitet mit Standardsoftware wie OPEN-ACCESS, SYMPHONY, KERMIT, CROSSTALK usw.
- Eingebaute Intelligenz erledigt Datentransfer mit jedem Computer oder Terminal über V-24-Schnittstelle, natürlich mit automatischer Baudratenerkennung
- Einfache Programmierung für Spezialaufgaben durch den Anwender
- Betrieb an Nebenstellenanlagen möglich
- Postzulassung mit FTZ-Nummer
- günstige Mengenstaffeln
- komfortables MAILBOX-Programm ebenfalls erhältlich

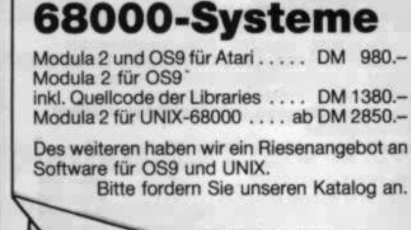
MICOM Computer- und Informationssysteme Entwicklungs- und Vertriebsges. mbH
Wüstenhofer Str. 6 · 5600 Wuppertal 1 · Tel. 02 02/44 34 01

MIELE-Datentechnik
Inh.: Hermann-Josef Miele
Bergfreiheit 60
5788 Winterberg-Silbach
Tel. (0 29 83) 83 07 u. 83 37

Modula 2 für 68000-Systeme

Modula 2 und OS9 für Atari . . . . . DM 980.–
Modula 2 für OS9˚
inkl. Quellcode der Libraries . . . . DM 1380.–
Modula 2 für UNIX-68000 . . . . ab DM 2850.–

Des weiteren haben wir ein Riesenangebot an Software für OS9 und UNIX.
Bitte fordern Sie unseren Katalog an.



Mehrplatz-Systeme mit OS-9 oder UNIX VMEbus oder ECBbus

USV-Anlage für Mikro- u. Minicomputer



Netzstabilisator mit 220 V Notstrom

- beseitigt Netzstörungen jeder Art
- schützt Daten und Programm
- 19"- oder Standgeräte
- sehr leise
- 200 VA/220 V ab **DM 909,72** (798,– + MwSt.)
- Modelle von 200 VA bis 60 kVA (220 V u. 380 V)

DVS Datentechnik GmbH
Ludwig-Thoma-Str. 1b · 8034 München-Germering
Telefon 0 89/8 41 90 64-66, Telex 5 24 966
Vertriebsbüro Rhein-Main
Breslauer Str. 21 · 6107 Reinheim bei Darmstadt
Telefon 0 61 62/33 14

RATEV Elektronik Vertriebs GmbH
Gothaer Straße 15, 4030 Ratingen 1
Telefon 0 21 02/4 20 51-52, Tx. 8 585 180 raf

286er-AT-Grundversion

Baby-AT-Board, XT-Format, Prozessor INTEL 80286, umschaltbar 6/8 (10) MHz, 0 Wait States, 1 MB RAM, bestückt mit 512 KB, Metall-Klapp-Gehäuse im Baby-AT-Format, drei Slim-line-Laufwerke möglich, Schlüsselschalter, LEDs für Power, Turbo Mode und Harddisk, Phoenix-BIOS, gepufferte Hardware-Uhr, 5× 16-Bit-Slot, 2× 8-Bit-Slot, Sockel für Co-Prozessor 80287, 180-W-Netzteil, parallele/serielle Schnittstelle (2. serielle optional), deutsche Tastatur mit 101 Tasten **DM 1850.–**

286er-AT-Standardversion

Grundgerät wie oben beschrieben, jedoch zusätzlich: Hercules-komp. Monochrome-Grafik-Karte mit zusätzlicher paralleler Schnittstelle, Western-Digital-Kombicontroller WD 1003 WA-2 (2× HD/2× FD), Mitsubishi-Floppylaufwerk MF 504 (360 K/1,2 MB) **DM 2498.–**

Zusätzliche Optionen

Aufpreise auf Standardversion:

20-MB-Festplatte, Fab. Seagate ST 225	**DM 545.–**
40-MB-Festplatte, Fab. Miniscribe MS 3650 (Slimline)	**DM 980.–**
EGA-Karte OEM 6, 640 × 350/480	**DM 378.–**
14"-Monitor, Fab. Addonics Dual Scan	**DM 290.–**
RAM-Aufrüstung auf 1 MB (512 K)	**DM 160.–**
Zweites Floppy-Laufwerk (1,2 MB)	**DM 249.–**
Zweites Floppy-Laufwerk (360 K)	**DM 198.–**
MS-DOS 3.2, deutsch, mit GW-Basic	**DM 215.–**

386er-AT-Grundversion

Baby-AT-Board, XT-Format, Prozessor INTEL 80386, 16-MHz-Taktfrequenz per Software umschaltbar, 0 Wait States, 2 MB RAM bestückt, bis 8 MB erweiterbar, 4× 8-Bit-Slot, 4× 16-Bit-Slot, ansonsten wie 286er-Grundversion **DM 4500.–**

286er-Baby-AT-Mainboard wie oben beschrieben	**DM 998.–**
386er-Baby-AT-Mainboard wie oben beschrieben	**DM 3800.–**
386er-Baby-AT-Mainboard, 20 MHz, 32 Bit geprüfter INTEL-Prozessor, 2 MB bestückt	**DM 4499.–**

Chipsätze von CHIPS + Technology



LAYTRONIC GmbH · Industriestr. 7, 7434 Riederich
Telefon 0 71 23/3 36 35, 3 42 23, Telex 7 245 409



LAYTRONIC GmbH · Industriestr. 7, 7434 Riederich
Telefon 0 71 23/3 36 35, Telex 7 245 409







mc-QUICKIE

mc-quickies sind aktuelle Produktanzeigen, mit denen Firmen ihre Produkte vorstellen. Verantwortlich für den Inhalt sind die Inserenten.



NEU
68020 – 68881
für
Selbstbaucomputer
c't 68000 GWK

Weltweit leistungsfähigstes Systemkonzept für modulare Systeme im Einfach-Europa-Format

Kostenloses Informationsmaterial anfordern!

Postfach 1360, D-5120 Herzogenrath,
Telefon (0 24 06) 60 35, Telex 8 32 109 gwk d

C-COMPILER

MI-C für CP/M, CP/M 86, MS DOS

Nutzen Sie die Vorteile von C
MI-C vereint hohen Bedienungskomfort mit hervorragender Leistung

- Vollständige Version mit 13stelliger BCD-Arithmetik für Gleitkommazahlen
- Erzeugt kurze und schnelle Programme, die auch in ein ROM gebraucht werden können
- Ausgabe in 8086- oder Z80-, 8080-Assemblercode
- Kompatibel zu MAC80/L80 (MASM/LINK) v. Microsoft
- Für 8086: 4 Speichermodelle/8087-Co-Prozessor
- Fehlerverfolgung mittels Trace möglich
- Umfangreiche Bibliothek (AMD9511-Paket erhältlich)
- UNIX-kompatibel
- Deutsche oder englische Version lieferbar 8"-/5,25"-/3,5"-/3"-Disk + dt. Handbuch

MI-C für CP/M	**445.– DM**
MI-C für CP/M 86 oder MS-DOS	**575.– DM**
MI-C Cross (Zielprozessor 8080/Z80)	**745.– DM**
MI-C Crossassembler + Linker (Ziel 8080/Z80)	**645.– DM**
MI-C Crosspaket (Ziel 8051)	**1495.– DM**

Herbert Rose, Bogenstraße 32, 4390 Gladbeck, Telefon 0 20 43/2 49 12 und 4 35 97
Vertrieb in Österreich:
Dr. Willibald Kraml, Microcomputer-Software, Degengasse 27/16, A-1160 Wien

---

DIN-A3-Plotter PLX-88

Spitzentechnologie zum Sensationspreis

Made in Germany

Preis DM
1498,-

- DIN-A3-Plotter
- 4 Farben m. autom. Stiftwechsel
- 0,025 mm Auflösung
- max. 200 mm/s Zeichengesch.
- HPGL-Kompatibel 28 Befehle
- Serienmäßig zwei Schnittst.
  Centronic u. RS 232 (V-24)
  ohne Aufpreis
- Ladbare Zeichensätze

Cmis GmbH
Gesellschaft f. Electronic u. Microprozessorsysteme mbH, Sebastianstraße 5, D-8480 Weiden, Telefon 0961/35053

SOFTWARE & HARDWARE
für Apple II & Mac & MS-DOS-Computer

z. B. für APPLE II und Kompatible

Wir liefern die RAMworks-Karte (AE) für den Apple IIe mit max. 16 MB (z. B. erweitert Appleworks) z. B. RAMWORKS III 64 K ...... DM 469.–
Anpassung für Appleworks 1.2 auf dem II+/e.
Original oder mit externer Tastatur. Anpassung in deutsch für SATURN 128 K und IBS AP33 1 MB! ...... DM 130.–
AD 16 Ch. 12 Bit, schnell! (Applied Eng.) ...... DM 799.–
PKASO/U-Printer-Karte (IS) ...... DM 429.–
Timemaster II H. O., **die** Uhrenkarte! (AE) ...... DM 299.–
ELF kompl. Statistik-Software (TWG) ...... DM 495.–
Prime-Plotter-Grafik-Software (Primesoft) ...... DM 629.–
Z-RAM Ultra 3, 256 K, Z80 CP/M + Uhr, max. 1 MB ...... DM 798.–
Z-RAM Ultra 2, 256 K, mit Uhr, max. 1 MB ...... DM 698.–

z. B. für IBM und Kompatible

MatchPoint-PC plus UNIFORM-PC (Lesen/Schreiben von Apple Disks im IBM PC/XT/AT/100 % komp. für DOS-ProDOS-CP/M) ...... DM 649.–
MatchMaker (Anschluß von Mac 800 K Drive an IBM PC/XT/AT/100 % komp. zum Datentransfer Mac-MSDOS) ...... DM 498.–
CompatiCard (Anschluß von 3.5", 5.25", 8" Drives an IBM PC/XT/AT/100 % komp. z. B. 3.5" 720 K/1.44 MB) ...... DM 629.–
XENO-PC (Cisin) (Lesen/Schreiben von CP/M- u. MS-DOS-Formaten im IBM PC/XT/AT/100 % komp.) ...... DM 258.–
ELF PC kompl. Statistik Software (TWG) ...... DM 500.–

z. B. für alle Systeme

Printerchanger 3 parall. Drucker auf 1 Micro inkl. Kabel/Netzteil (Keyzone) ...... DM 340.–
Printersharer 3 Micros auf 1 parall. Drucker inkl. Kabel/Netzteil (Keyzone) ...... DM 340.–

Wir sind Import-Spezialisten und bieten Ihnen eine große Auswahl an Software und Hardware bedeutender Hersteller aus den USA und England. Informationen gegen DM 3.– in Briefmarken.

WEISS COMPUTER **Dipl.-Psych. Karl-Heinz Weiß**
Am Wiesenhof 17, 2940 Wilhelmshaven, Tel. 0 44 21/8 31 79

Buchhaltung mit BUCH

Doppelte Buchführung, die auf Selbständige und Kleinunternehmer zugeschnitten ist:

- **14 Tage frei zur Ansicht**
- einfache und sichere Benutzerführung
- Prüfsiegel „anwenderfreundliche Software"
- Betriebsübersicht, G&V, Debitoren, Creditoren, Journal, Kontenblätter, Saldenliste, Bilanz usw.
- max. 32 000 Buchungen, 1200 Konten
- automatische Umsatzsteuerberechnung
- **Interface für DATEV, DBASE usw.**

... und so urteilt die Presse:

„einfach zu bedienendes Programm, das dem Einsteiger entgegenkommt" (CHIP 6/87)
„... ein leicht zu beherrschendes Finanzbuchhaltungsprogramm für kleinere Betriebe, das sehr anwenderfreundlich konzipiert wurde" (c't 7/87)
„... ein leistungsfähiges, leicht zu bedienendes Werkzeug" (PC + Soft 7/85)
„halten wir dieses Finanzbuchhaltungsprogramm für ganz ausgezeichnet" (Softwaretestjahrbuch 86)

180.– bis 680.– DM für PC/XT/AT, Macintosh, II+,c,e,gs

RÖNTGEN SOFTWARE
Oltmannsstr. 34, 7800 Freiburg, Tel. 07 61/40 87 40

Ecosoft Economy Software AG

Postfach 1905, D-7890 Waldshut, Tel. 0 77 51-79 20
Villa Domingo, CH-6981 Astano, Tel. 0 91-73 28 13

Prüf-Software und Frei-Programme (fast) gratis

Über 3000 Disketten mit professioneller Prüf-Software (Programme vor Anwender-Registrierung prüfen), aber auch nützlichen Amateur-Programmen für den beruflichen und privaten Gebrauch für IBM-PC/Kompatible, Macintosh, Atari ST, Amiga, C64/128, Apple II.

Katalog auf Disketten und Verzeichnis DM 10.–
(Bitte Computermodell angeben und Banknote oder Scheck beilegen.)

Neu: Emulation von Fremdsoftware, z. B. MS-DOS auf Amiga, C64 auf Amiga, Macintosh auf Atari ST usw. Gratis-Info-Schrift verlangen. 189

Gegen Einsendung dieses Inserates erhalten Sie zusätzlich einen Gutschein für eine Gratis-Ecosoft-Diskette.

Digital-oszilloskop

Bausatz ab
399.– DM

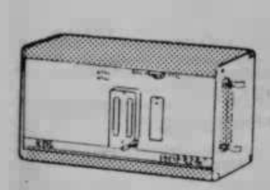
- für normales Fernsehgerät oder Monitor
- acht Oszillogramme im Speicher ablegbar
- max. Abtastfrequenz: 1 MHz
- Zeitbasis 10 µs/Teil...100 s/Teil
- erweiterbar auf 1 Std./Teil
- Ein- oder Zweikanalversion (aufrüstbar)
- eingeblendetes Raster
- alle Standard-Bedienfunktionen implementiert
- ideal auch für Schulen

Prospekt anfordern bei:

NEUCOM-ELECTRONIC GmbH
Hangweg 4, 8893 Hilgertshausen,
Telefon 0 82 50/14 25

Der Programmer: U N I P R O G !!

! Komfortabler EPROM-Programmer für PCs mit DOS ab 2.xx ! Das bewährte Programmiergerät mit professioneller Ausstattung ! Ideal für Werkstatt und Service ! Anschluß über Printerport, belegt keinen Slot im PC, automatische Printerport-Address-Ermittlung, keine Anpassung PC - **seitig notwendig.** Programmiert 2716 - 27256, auch C M O S und 25xx- Typen, natürlich F A S T - PROGRAMMING - Algorithmen ! Die komfortabel zu bedienende Treiber - SW ist vollständig menuegesteuert und unterstützt sowohl CGA als auch HERCULES - Karten.



Menuetexte und PROM-Spezifikationen sind benutzerseitig sehr leicht ergänzbar ! HEX- und ASCII-Daten-Darstellung mit komfortabler Editiermöglichkeit. Generiert und verarbeitet beliebige Dateien im INTEL-HEX-Format. Mehrfach industriell erprobt und dann ausschließlich verwendet!

DM 570,–

Ausführliche Information und Bezug durch: > > > > > > >
W - S Electronic, Augustusstr. 4, 8900 Augsburg 21

---

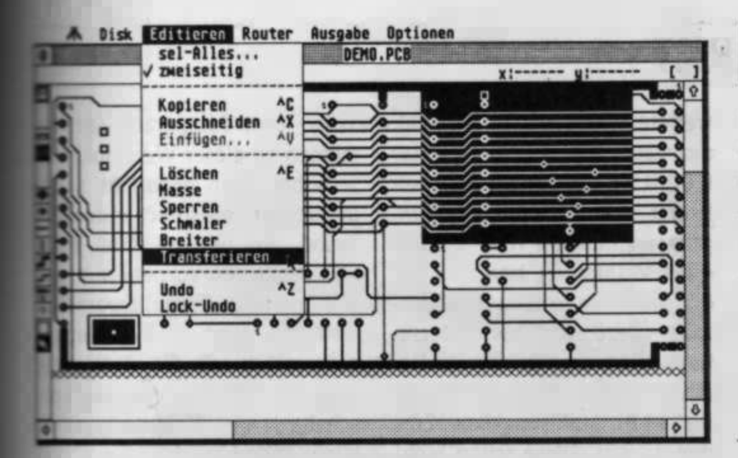
e d i c t a

Vertriebsgesellschaft für elektronische Bauelemente mbH
Löwenstr. 68, 7000 Stuttgart 70
Tel.: 0711/763381

68020RC20	595.-
68020RC16	495.-
68881RC16	495.-
EIZO8060S	1498.-
HDPLUS20	1298.-

Fordern Sie unsere **neue** Lagerliste 15/88 an.
Zwischenverkauf vorbehalten.

e d i c t a



So sieht ein Platinenentwurf mit PCB-layout auf dem Atari ST aus

Atari ST entwirft Leiterplatten

PCB-layout, so nennt sich ein interaktives Platinen-Entwurfsprogramm für den Atari ST, das mit der GEM-Benutzeroberfläche arbeitet und deshalb einfach zu bedienen ist. Es erlaubt den Entwurf von doppelseitigen Platinen bis zu einer Größe von 30 × 20 cm mit einer Auflösung von 1/20 Zoll.
Neben den Plazier-, Verschiebe- und Kopier-Funktionen kann man auch eine automatische Verdrahtungs-Routine aktivieren, die dann einzelne, herausgegriffene Leiterbahnen automatisch verlegt. Die Ausgabe des Layouts kann mit einem Epson FX-85 oder einem NEC-P6 bzw. einem HPGL-Plotter erfolgen. Für die jetzige Version sind noch unter 200 DM zu bezahlen, eine erweiterte Version mit komplettem Autorouter wird demnächst für 298 DM herauskommen.

Dipl.-Ing. Praefcke
Holzvogtkamp 55
2302 Flintbek
☎ 0 43 47/5 31

MSR mit Mira

Das modulare Meß-, Steuer- und Regelungs-Interface „Mira" als Bindeglied zwischen einem PC und Sensoren bzw. Stell-Elementen läßt sich über ein Druckerkabel am Computer anschließen. Eine zusätzliche Steckkarte ist nicht erforderlich, und mit der MS-DOS-Treibersoftware kann der Anwender in Basic, Fortran, Pascal oder in Assembler beliebige Routinen zur Meßdatenerfassung und Prozeßregelung schreiben.
Folgende Interface-Baugruppen können in dem 19-Zoll-Grundgerät installiert werden: Relaismodule, A/D- bzw. D/A-Wandler, digitale I/O-Module, Zähler bzw. Zeitgeber, Spitzenwertspeicher, Transientenrecorder-Module sowie Baugruppen zur Meßwerterfassung mit Sensorsignal-Vorverarbeitung.
Auf Wunsch wird das gesamte System mit einer Watchdog-Schaltung ausgerüstet, die bei Ausfall des Rechners oder der Verbindungsleitung alle Ausgangssignale in den sicheren Null-Zustand zurücksetzt.

Etewe
Schönfeldstr. 1
7500 Karlsruhe 1
☎ 07 21/69 55 56

Software zum Bürogebrauch

Für jeweils 398 DM sind bei CE-TEC jetzt fünf Bürosoftware-Pakete erhältlich: F & A Primus (Datenbank mit Textverarbeitung und Mailmerge), SuperCalc Primus (Tabellenkalkulation), SuperProject Primus (Projektmanagement mit der Netzplan-Methode), Graph in the Box (speicherresidentes Grafikprogramm) und Wordstar Publisher Primus (Text- und Adreßverwaltungs-Programm mit Grafikfunktionen). Die Programme laufen auf allen MS-DOS-Rechnern und sind im Prinzip leicht gekürzte Ausgaben bekannter Softwarepakete wie SuperCalc, Wordstar usw.. Eine durchdachte Benutzeroberfläche erleichtert dem Anwender das Einarbeiten – zusätzliche Fragen werden in den didaktisch gut gestalteten Handbüchern (in deutscher Sprache) beantwortet.

CE-TEC Trading
Kornkamp 4
2070 Ahrensburg
☎ 0 41 02/4 90 10

Hohe Auflösung mit 24 Nadeln

Zwischen Computer und einem 24-Nadel-Drucker eingeschleift werden die Interface-Geräte der Multicon-Serie. Mit ihnen ist dann – sofern ein Laserdrucker-Treiber im PC installiert ist – eine Druckausgabe mit maximal 360 Punkten/Zoll realisierbar. Es können aber auch Epson-9-Nadel-Treiber-Routinen verwendet werden. Das Grundgerät Multicon 24 kostet 480 DM, einen zusätzlichen 1-MByte-Speicher kann der Typ Multicon 1000 (1120 DM) anbieten. Beide Geräte sind einstellbar auf alle gängigen 24-Nadel-Drucker mit Centronics-Schnittstelle. Von der Auflösung her läßt sich damit schon „Laser-Qualität" erreichen, allerdings eben mit der Punktdicke der Matrix-Nadeln.

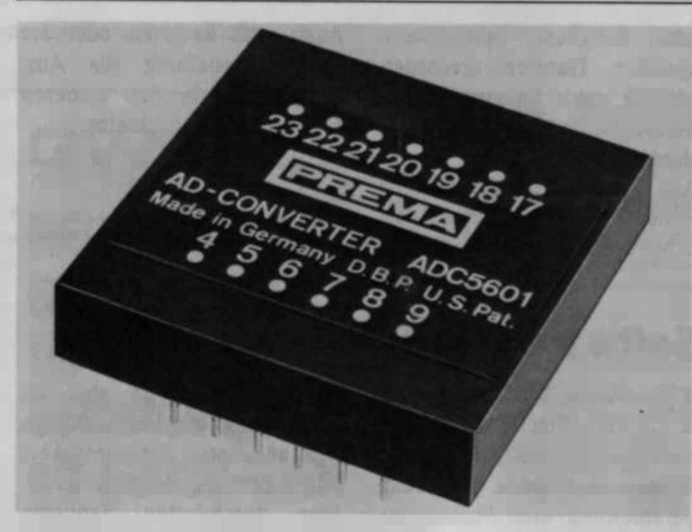
Fotronic
Josef-Beiser-Str. 13
8000 München 83
☎ 0 89/6 37 21 90



Individuell an Meß-, Steuer- und Regelaufgaben anpaßbar ist das PC-Interface-System „Mira"



Ausdrucke mit 300 bzw. 360 Punkten/Zoll erlaubt das Multicon-Interface für 24-Nadel-Drucker



Die volle Auflösung von 25 Bit erreicht der ADC 5601 nach 20 s

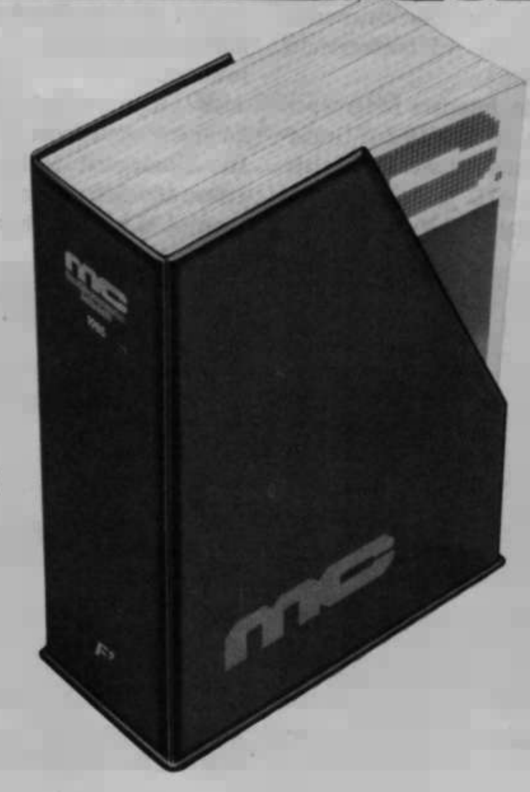
A/D-Wandler mit 25 Bit

Hauptsächlich für Präzisions-Anwendungen gedacht ist der ADC 5601, ein 25-Bit-Wandler-Modul, das drei Versorgungsspannungen braucht. Der Hybrid-Baustein arbeitet nach einem patentierten Mehrfach-Rampen-Verfahren. Prinzipiell wird die Art seines Einsatzes durch die externe Beschaltung festgelegt. Neben der genannten hohen Auflösung sind weitere wichtige Daten ein Linearitätsfehler von unter 0,0001 %, ein Temperaturkoeffizient von 0,5 ppm/K, Eingangsspannungen von −2,5 bis +2,5 V (bipolar) bzw. −5 bis 0 V sowie die serielle Datenausgabe. Bezüglich der Integrationszeiten spezifiziert das Datenblatt einen Wert von 10 ms (bei 14 Bit Auflösung) bzw. 20 s (bei 25 Bit). Diese schon recht lange werdende Integrationszeit muß natürlich bei der Systemauslegung berücksichtigt werden, doch z. B. in manchen Wäge- oder Temperatur-Meßsystemen ist dieser Wert noch tolerierbar, man bekommt dafür ja auch die entsprechende Präzision. Im übrigen muß man sich nicht von vorneherein auf bestimmte Einstellparameter festlegen, denn die Wandlungszeit und damit die Auflösung sind per Software vom nachgeschalteten Prozessorsystem leicht änderbar.

Prema
Robert-Koch-Str. 10
6500 Mainz 42
☎ 0 61 31/5 06 20

24-Nadeln für unter 1000 DM

Als Preis-Leistungs-Genie bezeichnet NEC seinen ersten 24-Nadel-Drucker, der weniger als 1000 DM kostet. Der Pinwriter P2200 erreicht beim Grafikdruck eine Auflösung von 360 × 360 Punkten pro Zoll und bietet für Textdrucke sechs Schrifttypen, darunter auch OCR-B. Über Wechselkassetten sind zwölf weitere Schriftarten verfügbar. Der P2200 bietet einen Speicher von 8 KByte und eine Reihe von Papierzuführungen. Standardmäßig eingebaut sind Schub- und Zugtracktor für





Bewährte NEC-Qualität präsentiert der 24-Nadel-Drucker P2200 für 998 DM

Endlospapier und ein halbautomatischer Einzelblatteinzug. Eine Papierzuführung am vorderen Teil des Druckers erlaubt den Einsatz von Einzelblättern, ohne das Endlospapier herauszunehmen. Als Option steht ein vollautomatischer Einzelblatteinzug zur Verfügung. Durch seine volle Kompatibilität zu den NEC-Pinwritern der 24-Nadel-Serie harmoniert der P2200 mit allen wichtigen Software-Paketen. Im Schnelldruck schafft der P2200 max. 168 Zeichen in der Sekunde, bei Letter-Quality noch 47 Zeichen in der Sekunde. Der Drucker ist serienmäßig mit einer parallelen Schnittstelle versehen; eine serielle Schnittstelle gibt es natürlich auch. Wem die Geräuschentwicklung von 57 dB(A) zu laut ist, kann auf den Quiet-Mode mit 54 dB(A) umschalten. Wir haben in der Redaktion einen P2200 über mehrere Wochen eingesetzt. Dabei hat sich der Drucker sowohl an einem Atari ST als auch an einem MS-DOS-Computer bestens bewährt. Mehrere Hardcopys, die mit dem P6-Treiber auf dem Atari ST angefertigt wurden, bestätigten die von NEC zu erwartenden Qualitätsansprüche. Der P2200 ist servicefreundlich. So kann z. B. der Druckkopf mit wenigen Handgriffen gewechselt werden.

NEC Deutschland
Klausenburger Str. 4
8000 München 80
☎ 0 89/93 00 60

PC-Softwareführer

In einer überarbeiteten Neuauflage ist der 736 Seiten umfassende „Softwareführer für PCs" erschienen. Er bietet für 34 DM eine gute Übersicht über das deutschsprachige Softwareangebot, und zwar mit über 3000 Programmbeschreibungen von rund 500 Anbietern, die komplett mit Anschrift verzeichnet sind.

Die Unterteilung erfolgte nach branchenunabhängigen, branchenspezifischen sowie technisch-wissenschaftlichen Programmen. Hinzu kommen Kapitel über Desktop-Publishing, Systemsoftware, Lernprogramme und Spiele.

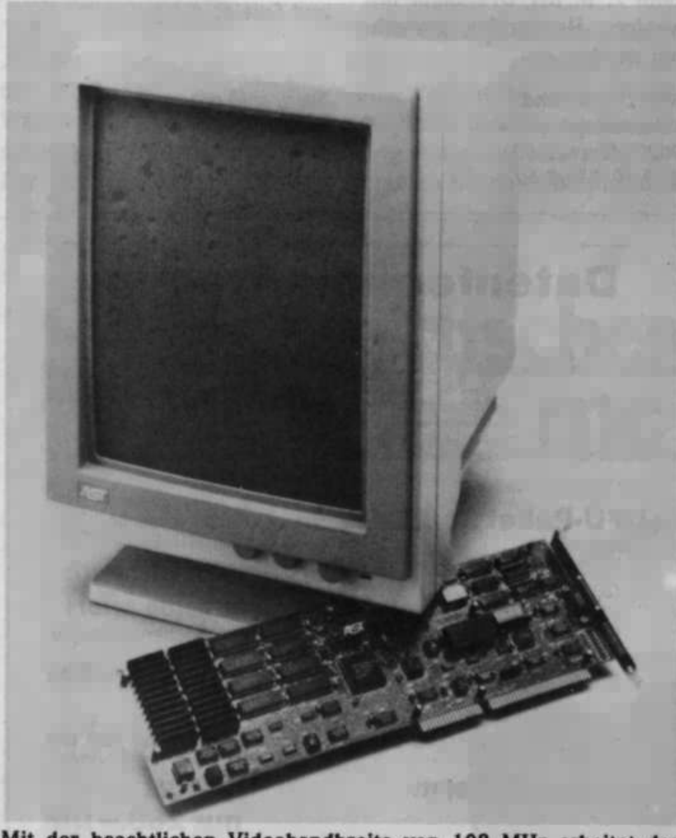
Rossipaul-Verlag
Bavariaring 24
8000 München 2
☎ 0 89/5 30 92 27

Schwarz auf weiß mit hoher Auflösung

In Desktop-Publishing-Computersystemen sind Ganzseiten-Monitore mit schwarzer Schriftdarstellung auf weißem Hintergrund unabdingbare Voraussetzung für optimales Arbeiten.

Ein Bildschirm aus dieser Kategorie ist der AST-TurboVision, der mit einer Video-Bandbreite von über 100 MHz arbeitet und es so auf eine Auflösung von 1024 Bildpunkten in 1280 Zeilen bringt. Mitgeliefert wird eine XT/AT-kompatible Grafikkarte mit dem Spezialprozessor 34010 von Texas Instruments. Der Systemtakt kann bis zu 12,5 MHz betragen, ein eventuell im PC verwendeter Farbgrafik-Adapter wird in der Ganzseiten-Betriebsart ignoriert und kann demzufolge im Slot verbleiben. Ein eigener Hercules-Modus sorgt dafür, daß praktisch jede denkbare Anwendungssoftware mit dem für knapp über 6000 DM erhältlichen Monitor zusammenspielt.

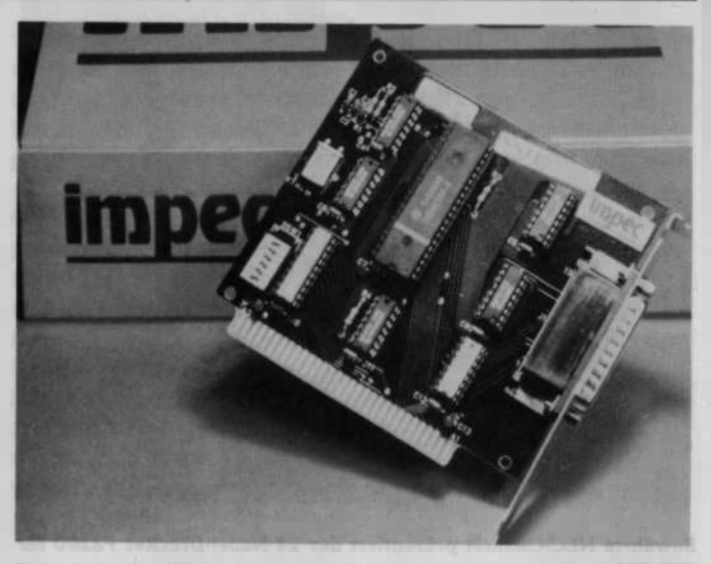
AST GmbH
Emmanuel-Leutze-Str. 1b
4000 Düsseldorf 11
☎ 02 11/5 95 70



Mit der beachtlichen Videobandbreite von 108 MHz arbeitet der Ganzseiten-Monitor „TurboVision"

Praxisgerechte Schnittstellenkarten

Zwei neue Interface-Platinen hat Impec in sein Vertriebsprogramm aufgenommen. Zum einen die IEC-Bus-Schnittstellenkarte (IEEE 488) mit DMA-Zugriff auf drei verschiedenen Kanälen. Zentrales Bauelement ist hier der Controller-Chip NEC D7210C. Zur Auslieferung kommt die Karte für 700 DM komplett mit Kabelsatz und englischer Beschreibung.



Distanzen bis 1000 m kann man mit der RS422-Schnittstellenkarte überbrücken

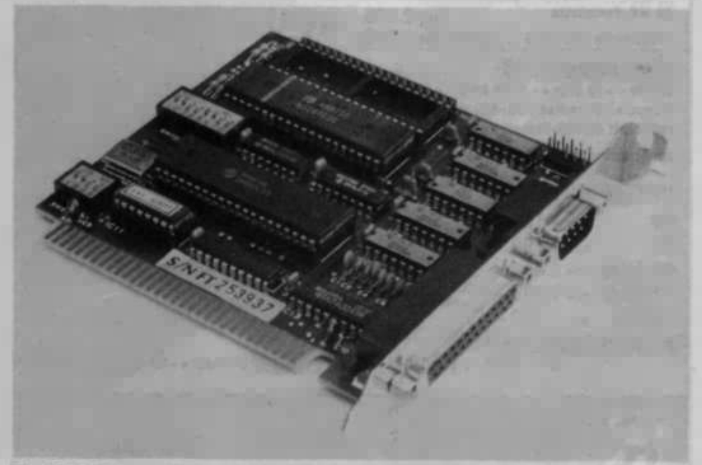
Ein serielles Interface nach dem EIA-422-Standard bietet die Platine mit dem Namen „RS 422". Das Kommunikations-Protokoll ist auf asynchronen Betrieb bis maximal 56 Bd ausgelegt, die anschließbaren Kabellängen können bis zu 1000 m betragen. Auch diese Karte (280 DM) ist wie die IEC-Bus-Platine als Zusatz für jeden MS-DOS-Computer geeignet.

Impec
Waldhörnlestr. 18
7400 Tübingen
☎ 0 70 71/7 00 20

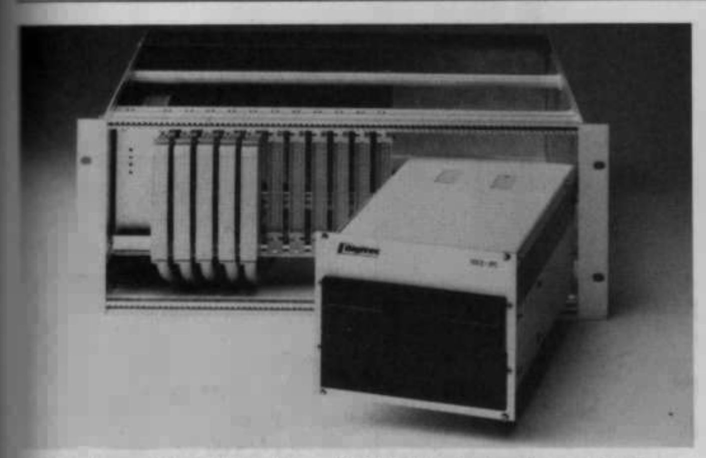
Seriell/Parallel-Karte

Für 135 DM ist die mit einer parallelen und einer seriellen Schnittstelle ausgerüstete PAL-Printer-Karte erhältlich. Besonders erwähnenswert an dieser kurzen PC-Einsteckplatine ist, daß eine zweite serielle Schnittstelle durch Ergänzung eines 16450-Bausteins eingerichtet werden kann. Deren Leitungen sind dann auf eine Pfostenleiste im Platinenbereich herausgeführt, während die beiden in der Grundausführung schon vorhandenen Schnittstellen an einen 9poligen D-Stecker (seriell) bzw. eine 25polige D-Buchse (Centronics) gelegt sind.

Paladin
Schlegelstr. 34
4019 Monheim 2
☎ 0 21 73/6 57 04



Die PAL-Printer-Karte kann man um eine zweite serielle Schnittstelle erweitern



Industriegerecht stabil in einem 19-Zoll-Einschub aufgebaut ist das XT-System von Digitec

Industriecomputer als Kassette

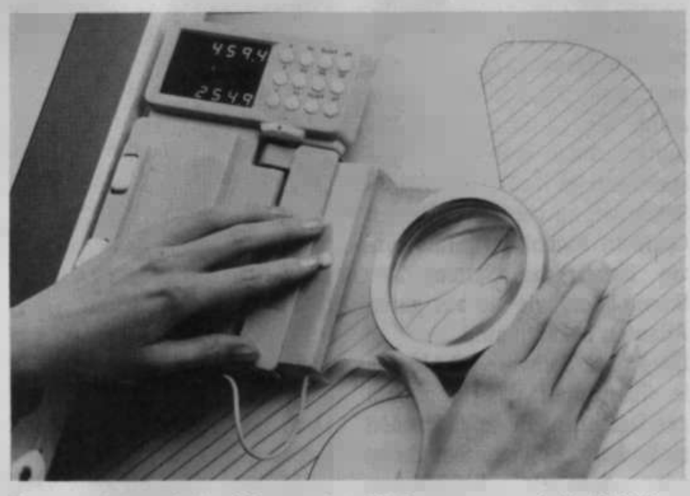
Optimiert auf Steuerungs- und Automatisierungs-Anwendungen ist das Industrie-PC-System ISS2-IPC, bei dem ein kompakter, XT-kompatibler Rechner in einer 19-Zoll-Einschubkassette untergebracht ist. In dieser Kassette befinden sich neben dem CMOS-Rechnerboard einschließlich Grafikkarte (Hercules oder CGA) noch zwei Floppy- bzw. – in einer anderen Ausführung – ein Floppy- und ein Festplatten-Laufwerk. Betriebssystem ist MS-DOS 3.2, über den ECB-Erweiterungsbus können gängige I/O-Baugruppen angesteuert werden. Als Optionen gibt es verschiedene Monitore, eine Tastatur, Optokoppler-Zusätze für die Tastatur, Schnittstellen-Erweiterungen sowie verschiedene ECB-Karten mit Slave-Prozessor-Aufsatz.

Digitec Engineering
Grünstr. 36
4005 Meerbusch 1
☏ 0 21 05/7 30 05

Zwei KByte in der Uhr

Mit einem Datenspeicher, der die Informationsmenge von rund einer Schreibmaschinenseite (2 KByte) aufnimmt, ist die Armbanduhr „Seiko Datagraph RC 4000" ausgerüstet. Damit hat der Benutzer wichtige Telefonnummern, Termine oder auch Preislisten jederzeit auf Abruf bereit. Der Clou an dieser Armbanduhr, die 299 DM kostet, ist ein kleiner Stecker, der den internen Speicher zum Einschreiben der Daten mit einem C-64 oder IBM-PC verbindet. Mit im Lieferumfang enthalten ist deshalb eine auf den jeweiligen Computertyp zugeschnittene Software-Treiberdiskette.

Intraprojects
Raphaelsweg 17
4047 Dormagen 1
☏ 0 21 06/4 10 57



Ein optoelektronisches Wegstrecken-Meßsystem sorgt beim DIGIS-Digitalisierer für ein exaktes Übertragen beliebiger Zeichnungen in CAD-Entwürfe

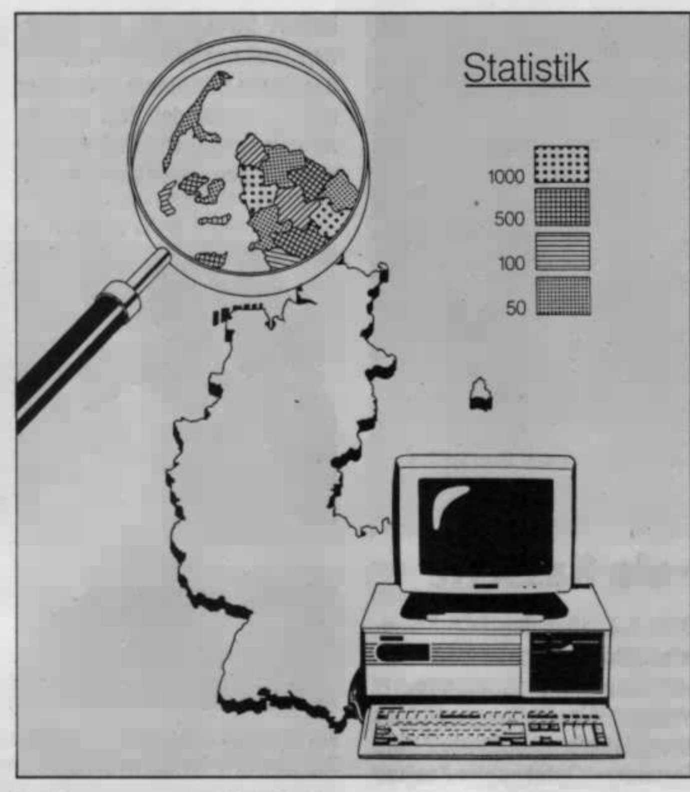
CAD mit dem Reißbrett

Eine schnelle Digitalisierung von beliebigen Zeichnungs-Vorlagen – um sie z. B. in CAD-Entwürfe einzubinden – erlaubt das Digitalisiersystem DIGIS. Die aufzunehmende Vorlage wird zunächst auf das Digitalisierbrett gespannt. Der Anwender braucht nur noch mit der Fadenkreuzlupe den Koordinatenlinien oder Begrenzungen des gezeichneten Entwurfs zu folgen, alles andere besorgt das optoelektronische Wegstrecken-Meßsystem. Erhältlich ist DIGIS mit Fadenkreuzlupe und Read-Taste oder mit Zeichenkopf, Linealen und aufsteckbarem Fadenkreuz. Vorhanden bei beiden Varianten ist jedoch die kleine Funktionstastatur, die der Steuerung des CAD-Systems vom Digitizer aus dient. Unmittelbarer Anschluß besteht zu den Autocad- und PC-Draft-Programmpaketen.

Franz Kuhlmann
Postfach 7 20
2940 Wilhelmshaven
☏ 0 44 21/18 11

Landkarte ersetzt Tabelle

Überall dort, wo umfangreiches Zahlenmaterial ganz bestimmten örtlichen Begrenzungen oder Regionen zugeordnet werden muß, könnte das PC-Softwarepaket MAP interessant sein. Das Namenskürzel steht für „Markt-Analyse-Planung“. Daten werden mit gespeichertem Kartenmaterial in Sekunden zu gebietsbezogenen Grafiken kombiniert. Im Prinzip liefert das System also Schaubilder, wie sie aus guten Atlanten bekannt sind.
Über eine Software-Schnittstelle übernimmt das Programm aus Datenbanken – das können z. B. Lotus, dBase, Symphony oder andere sein – die erforderlichen Daten. Die Ausgabe erfolgt in bis zu 16 Farben und 300 Mustern bzw. Schraffuren. Für Druckvorlagen empfiehlt sich dabei die Erstellung von Dias mit einem speziellen Belichtungsgerät, für Präsentationen der Ausdruck auf Overhead-Folie, für alle anderen Bereiche genügt unter Umständen schon ein Drucker bzw. Plotter. Als Computer eignet sich jeder XT/AT mit entsprechender Grafikausrüstung.

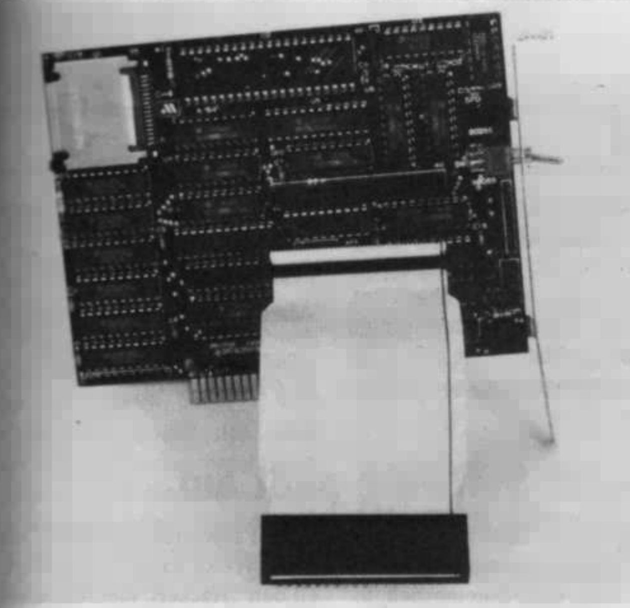


Das Programmpaket MAP faßt Regional-Zahlenmaterial zu anschaulichen Karten-Darstellungen zusammen

IBS
Postfach 10 02 43
3016 Seelze
☎ 05 11/4 00 20

PC im AT-Tempo

Mit einer kurzen Steckkarte kann man seinem PC eine 80286-CPU mit der entsprechenden AT-Geschwindigkeit spendieren. Zu diesem „Tuning“ wird die vorhandene CPU aus der Mutterplatine entfernt und in die Beschleunigerkarte gesteckt. Gleiches ge-

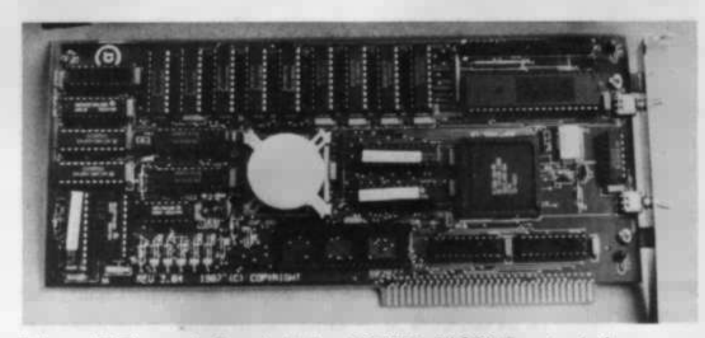


Eine kurze Platine bringt den 80286-Takt in den PC

schieht mit einem eventuell vorhandenen Mathematik-Coprozessor. In den nunmehr verwaisten CPU-Platz der Hauptplatine kommt der Zusatzkarten-Verbindungsstecker. Sollten einige wenige Programme (meistens Spiele) mit dem neuen 5- oder 8-MHz-Takt des 80286 nicht zurecht kommen, kann man immer noch per Schalter den alten 8088 in Betrieb setzen.

Nach Angaben der Herstellerfirma führt die Beschleunigerkarte zu einer Steigerung der Verarbeitungsgeschwindigkeit um den Faktor 2,2. Aktiviert man dann noch einen Teil des Arbeitsspeichers als Cache-Memory (über Steckbrücken einstellbar), soll sogar der Faktor 6,6 erreicht werden – das Ganze zu einem Endpreis von unter 700 DM.

SCSE Computer Systeme
Haselmühlweg 15A
8750 Aschaffenburg
☎ 0 60 21/4 30 40



Bringt AT-Eigenschaften in jeden PC/XT: 80286-Zusatzplatine

286-CPU zum Nachrüsten

Als Einsteck-Platine für IBM-PCs/XTs gibt es jetzt eine 80286-Option mit 10 MHz Taktfrequenz. Mit auf dem Board sind 8 KByte Cache-Memory sowie eine Fassung für den 80287. Die Platine ist voll DMA-tauglich und arbeitet asynchron mit dem vorhandenen 8088.

Kitec
Postfach 82 04 05
8000 München 82
☎ 0 89/4 39 17 08



Wer „aufgeladen" ist, sollte auf das Knöpfchen des StaticClear-Kästchens drücken

Antistatisches auf Knopfdruck

Elektrostatische Aufladungen anzeigen und sie gleichzeitig beseitigen – das kann „Misco StaticClear", ein kleines Kästchen, das in der Nähe des DV-Arbeitsplatzes aufgestellt werden sollte. Wenn z. B. ein Computer-Benutzer gerade über einen Kunststoff-Teppich gehen mußte und deshalb eine Aufladung zu befürchten ist, tippt er lediglich mit dem Finger auf die kleine Taste des Gerätes – ein darüberliegendes Display zeigt im selben Moment an, ob eine Aufladung vorhanden ist. Da die Drucktaste eine elektrisch leitende Oberfläche besitzt, wird die statische Elektrizität gleich abgeleitet. Neben dieser interessanten Vorbeuge-Maßnahme gibt es im Produktprogramm von Misco auch diverse Antistatik-Teppiche oder -Matten.

Misco
Nordendstr. 72/74
6082 Mörfelden-Walldorf
☎ 0 61 05/40 10

Codier-Software paßt auf

Mit dem Datenverschlüsselungs-Programm „Proficode" kann man Dateien beliebigen Formats aufgrund einer individuellen Paßwort-Eingabe ver- bzw. entschlüsseln. Das wäre eigentlich noch nichts besonderes, aber der Clou an diesem Programmpaket ist die „Aufpasser-Funktion": Versucht ein Unbefugter, mit der Eingabe allgemein üblicher Paßwörter an den Ursprungs-Text heranzukommen, so werden diese Fehl-Eingaben gespeichert und dem berechtigten Benutzer nach Eintippen des richtigen Paßwortes mitgeteilt. Auf diese Weise kann man eventuell den „Hacker" identifizieren. Der Anwender allerdings, der sein eigenes, korrektes Paßwort vergißt, kommt auch nicht mehr an seine Originaldatei heran – die Schlüsselcodierung ist aus keinem Programmteil zu rekonstruieren und kann auch vom Software-Lieferanten nicht „herausoperiert" werden.

iM Software
Postfach 70 22
7250 Leonberg-Warmbronn
☎ 0 71 52/7 10 93

Stromsparendes CMOS-PLD

Valvos erste programmierbare Logikschaltung in CMOS-Technik kommt jetzt unter der Bezeichnung PLC 473 heraus. Es handelt sich dabei um ein elektrisch programmierbares und UV-löschbares Logik-Array, bestehend aus 24 Und- sowie 22 Oder-Gattern in einem 24poligen DIL-Gehäuse. Die Programmierung kann beispielsweise mit dem Softwarepaket „Amaze" erfolgen, es läuft auf IBM-PCs oder VAX-Maschinen. Die insgesamt 20 I/O-Anschlüsse kann der Anwender wahlweise als Ein- oder Ausgänge definieren.

Valvo
Burchardstr. 19
2000 Hamburg 1
☎ 0 40/3 29 60

Brain – die intelligente Datenbank

Eine Menge angenehmer Eigenschaften besitzt die – natürlich nicht ausschließlich – zur Archivierung von Literaturquellen entwickelte Datenbank „Brain". So paßt sich z. B. die Länge des jeweils benötigten Feldes exakt der Eingabe an (bis zu einer halben Schreibmaschinenseite), was natürlich den beanspruchten Speicherplatz optimal ausnutzt. Die Eingabe selbst erfolgt mit komfortablen Masken; drei sind schon vorgegeben, weitere lassen sich beliebig selbst erstellen, und zwar ohne Kenntnisse einer Datenbanksprache. Häufig benötigte Wörter (z. B. lange Zeitschriftentitel) sind per Tastendruck abrufbar.

Das alphabetische Sortieren erfolgt automatisch, vom Anwender ist nur das jeweilige Feld zu bestimmen. Beim Ausdruck kann der Benutzer entscheiden, ob z. B. Vornamen nur als Initialen, die Jahreszahl in Klammern, der Titel nur bei Büchern unterstrichen, der Verlag vor dem Autor oder umgekehrt zu Papier gebracht werden soll. Das MS-DOS-Paket kostet 425 DM; eine Demo-Diskette gibt es für 20 DM.

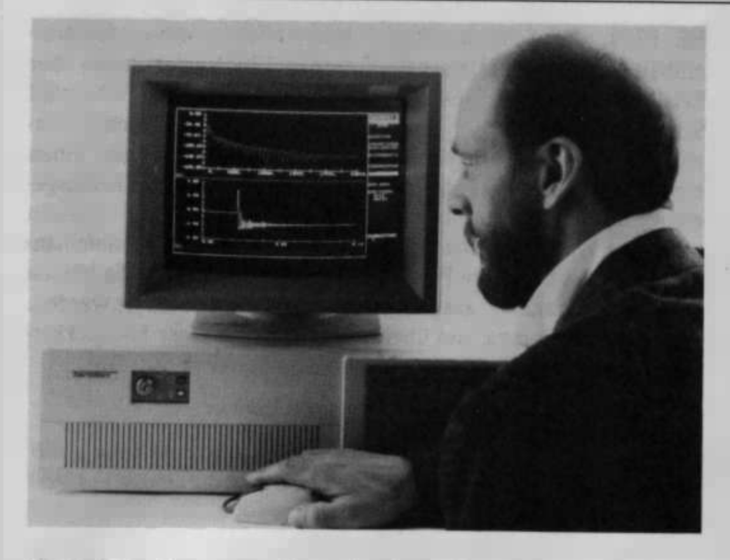
Shamrock Software Vertrieb
Karlstr. 35
8000 München 2
☎ 0 89/59 54 68

Antriebe vom PC aus steuern

Über zwei unabhängige, bipolare Analogausgänge mit 11 Bit Amplituden-Auflösung verfügt die PC-Einsteckplatine HE 8771. Als Schaltausgänge enthält die Karte fünf Umschalt-Relais sowie zwei Öffner/Schließer-Kontaktsätze. Sechs von diesen Relais können mit einem PIA-Baustein angesteuert werden, das siebte über eine eigene Watchdog-Schaltung. An Eingängen bietet die Platine acht Digital-Kanäle, ausgelegt für 24-V-Ansteuerung und über Optokoppler galvanisch getrennt.

Zur Versorgung dieser hauptsächlich als Antriebs-Controller verwendeten Baugruppe kann man entweder die computerinternen 12-V-Versorgungen oder ein externes ±15-V-Netzteil heranziehen. Sämtliche Ein- und Ausgänge sind auf einen 37poligen D-Stecker herausgeführt, softwareseitig müssen vom Anwender acht aufeinanderfolgende Ansprechadressen (frei wählbar) definiert werden. Der Preis für diese Einsteckkarte liegt bei etwa 500 DM.

Hesch-Schröder
Boschstr. 8
3057 Neustadt 1
☎ 0 50 32/10 21



Komfortable Frequenzanalyse mit einem Erweiterungs-Softwarepaket des Grundprogrammes „Signalys"

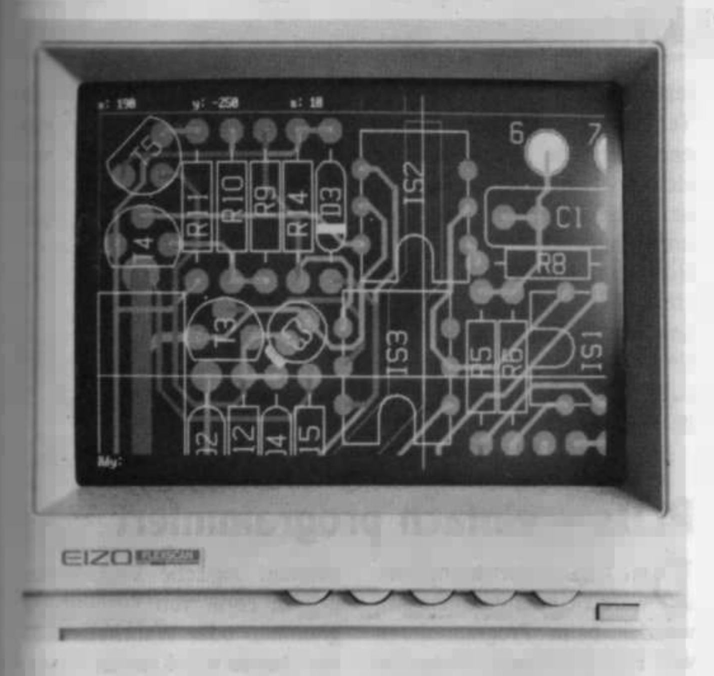
Saubere Frequenzanalyse

Das von Ziegler unlängst vorgestellte Software-Paket Signalys wurde jetzt um den Zusatz „Frequenzanalyse" erweitert. Dieses Modul enthält neben Algorithmen zur Fast-Fourier-Transformation auch leistungsfähige Funktionen zur digitalen Filterung, spektralen Glättung, zum Errechnen diverser Spektren und Korrelations-Zusammenhänge. Genau wie das Grundpaket ist auch diese Zusatz-Software menügeführt (in deutscher Sprache), so daß sich die Anwender in kurzer Zeit einarbeiten können. Das Programm wurde auf nahezu 25 unterschiedlichen, IBM-kompatiblen Rechnern getestet und arbeitet mit 40 verschiedenen Grafik-Controllern zusammen. Ausgeliefert wird es mit ausführlicher Dokumentation und Schulungs-Unterlagen.

Ziegler Instruments
Postfach 20 14 65
4050 Mönchengladbach 2
☎ 0 21 66/8 68 10

Platinen-Entwurf in 16 Ebenen

Zum Preis von unter 5000 DM ist bei Walter Electronic das Programmpaket „Draftsman-EE" erhältlich, eine Leiterplatten-Entwurf-Software, die mit einer Auflösung von 0,0254 mm (1 mil) arbeitet. Grundbestandteil des Paketes ist ein leistungsfähiger Grafikeditor, mit dem Stromlaufpläne, Platinen-Entwürfe sowie technische Zeichnungen erstellt werden können. Zu bedienen ist das Ganze entweder via Tastatur oder Maus, es stehen dabei 16 frei benutzbare Datenebenen zur Verfügung. Treiber für die Ausgabe auf Plottern oder Matrixdruckern sind im Programm vorhanden.
Ein vollkommener Design-Arbeitsplatz wird das Ganze schließlich mit dem separat erhältlichen „Autorouter", der bis



Wegen der hohen Entwurfs-Auflösung arbeiten das Programmpaket „Draftsman-EE“ und der Autorouter-Zusatz idealerweise auf einem EGA-Computer

zu 16 Multilayer-Ebenen einer Leiterplatte automatisch verdrahten kann. Das Interessante an diesem Zusatzprogramm ist der ständig auf dem Bildschirm sichtbare Arbeitsverlauf, so daß man gegebenenfalls noch stoppen und etwas hinzufügen kann. Die genannten Programmpakete laufen auf IBM-PCs ab der Kategorie XT mit 10 MByte freier Festplatten-Kapazität. An Grafik-Standards werden CGA, Herkules und EGA unterstützt.

Walter Electronic
Lindenstr. 3
7901 Schnürpflingen
☎ 0 73 46/89 10

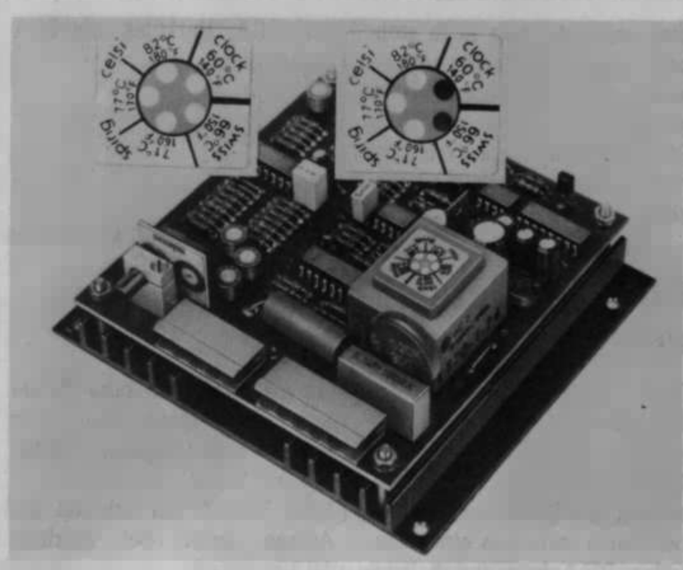
Computer „telefonieren“ miteinander

Mit dem Data Carrier System DCS 3 von Teltone kann der Anwender bestehende Telefonleitungen als einen unabhängigen Daten-Kommunikationskanal zwischen einer zentralen DV-Anlage und beliebigen Terminals, PCs, Druckern usw. nutzen.

Die Anlage setzt dabei alle Daten auf eine Trägerfrequenz von 40 kHz bzw. 80 kHz um und erlaubt so auf beiden Frequenzen gleichzeitiges Senden und Empfangen, noch dazu mit dem Vorteil, daß die normale Benutzung der Telefonleitung für die Sprachübertragung wegen der hohen Trägerfrequenzen unbeeinflußt bleibt.

Vom Prinzip her arbeitet die Anlage seriell bei maximal 19 200 Bd. Die nur aus wenigen Baugruppen zusammengesetzte Übertragungseinrichtung kann innerhalb kurzer Zeit in jede beliebige Nebenstellenanlage eingebaut werden.

Atlantik Elektronik
Fraunhoferstr. 11A
8033 Martinsried
☎ 0 89/8 57 00 00



Abgestuft nach verschiedenen Temperaturbereichen gibt es die Celsiclock-Meß-Etiketten

Aufkleber messen Temperaturen

Hinter der Bezeichnung „Celsiclock" verbergen sich vielseitig anwendbare Temperatur-Meß-Etiketten, auf denen verschiedene temperaturempfindliche Zonen kreisförmig angeordnet sind. Damit läßt sich im Bereich von 40...260 °C (in praxisgerechten Abstufungen) die jeweils zuletzt vorhandene Maximal-Temperatur bestimmen: Der entsprechende Farbpunkt färbt sich dauerhaft schwarz. Auf diese Weise kann man auch an unzugänglichen oder unter Hochspannung stehenden Flächen recht bequem Maximal-Temperaturen bestimmen.
Vom gleichen Hersteller kommt auch hitzeempfindlicher Lack in verschiedenen Temperatur-Abstufungen auf den Markt. Irgendwo mit einem kleinen Pinsel aufgetupft, zeigt er das Überschreiten der jeweiligen Temperatur durch dauerhaften Farbumschlag an.

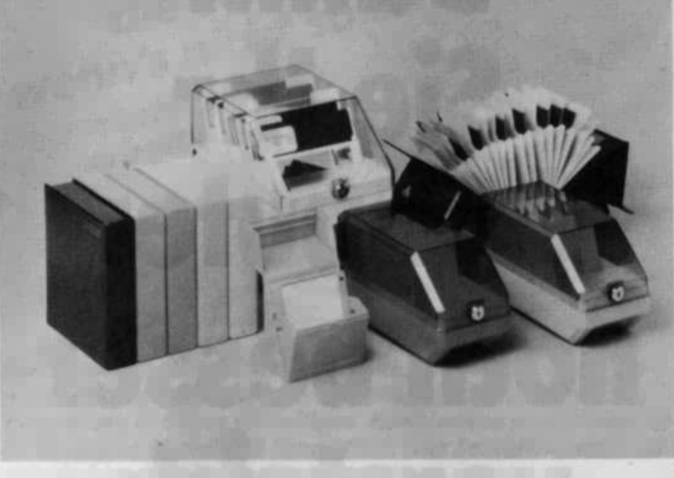
Cobonic
Postfach 11 38
7737 Bad Dürrheim
☎ 0 77 26/14 90

PLDs – einfach programmiert

Die Logik-Entwicklungssoftware CUPL bietet dem Anwender beim Programmieren von PLD-Bausteinen (Programmable Logic Devices) eine sehr komfortable Arbeitsumgebung. So braucht sich der Entwickler z. B. bei der Formulierung der logischen Verknüpfungen weder um die Definition der H- bzw. L-Pegel noch um ein bestimmtes Eingabeformat zu kümmern. Das Softwarepaket versteht logische Verknüpfungen in Form von Zustandsdiagrammen oder Wahrheitstabellen ebenso wie Klartext (Hochsprachen).
Derzeit können rund 100 verschiedene PLD-Chips von insgesamt 15 Herstellern programmiert werden, zukünftige Bausteine lassen sich der internen Bibliothek ohne großen Aufwand hinzufügen. Die Software kann auch leicht in größere Pro-



Zur PLD-Entwicklungs-Software CUPL gibt es auch verschiedene Programmiergeräte



Ordentlich und gut geschützt aufbewahrt sind Disketten aller Größen in diesen farbigen Boxen

grammpakete eingebunden werden, z. B. in Schaltungs-Entwicklungssoftware wie „p-cad". Lauffähig ist das Programm auf jedem MS-DOS-System, spezielle CUPL-Versionen sind auch für Unix-Workstations und für VAX-Rechner (unter Unix oder VMS) verfügbar. Der Preis liegt bei knapp unter 3000 DM. Das sind pro Chip-Typ nicht mehr als 10 DM.

Neumüller
Eschenstr. 2
8028 Taufkirchen
☎ 0 89/61 20 80

Schutz für Hard- und Software

Gegen lästige und unschöne Staubschichten auf Computern, Druckern und Tastaturen helfen die Boeder-Schutzhauben aus pflegeleichtem Kunstleder. Die Besonderheit ist ein weiches Innenfutter, das die abgedeckten Geräteoberflächen zusätzlich schützt. Mittlerweile gibt es eine Vielzahl verschiedener Größen, so daß alle gängigen Computer, Monitore, Ta-

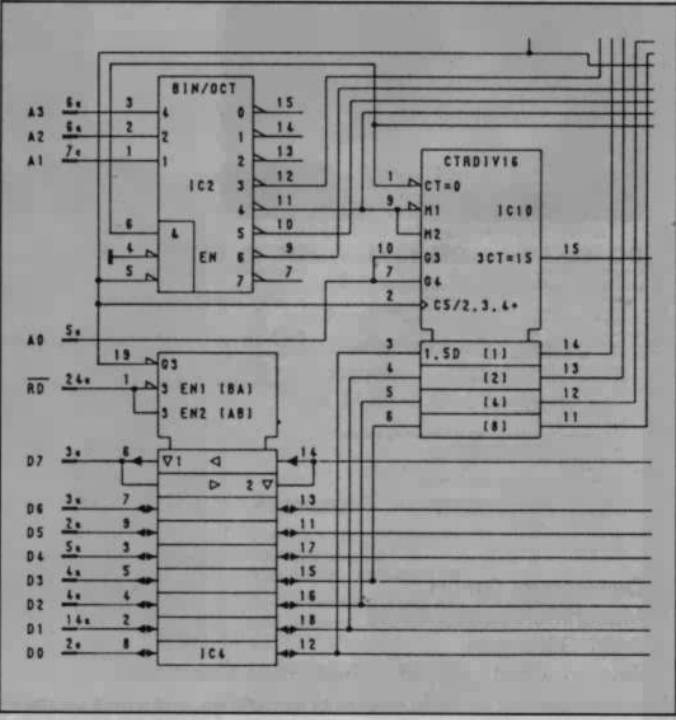
staturen und Drucker „unter die Haube" zu bringen sind. Für schwierige Fälle gibt es Universal-Modelle.
Ein weiteres nützliches Zubehör sind Klarsicht-Diskettenboxen und Floppy-Ablagebehälter. Sie gibt es in verschiedenen Grundfarben und für alle üblichen Disketten-Größen, so daß man an der Farbgebung die Zuordnung zu verschiedenen Arbeitsbereichen erkennen kann. Die Arbeitsboxen mit Klarsichtdeckel sind abschließbar und erlauben das übersichtliche Sortieren der Floppys wie in einer Registratur mit Trenn-Abteilen.

Döbbelin & Boeder
Wickerer Str. 50
6093 Flörsheim
☎ 0 61 45/50 20

Schaltplan-Zeichnen

Für unter 500 DM ist das Zeichenprogramm „Shamrock-CAD" zu haben, das auf jedem PC mit mindestens 256 KByte RAM läuft. Zum Entwerfen von Grafiken (mit Maus oder Tastatur) stehen die Grundelemente Linie, Kreis und Kreissegment sowie ergänzend die Texteingabe zur Verfügung. Mehrere Bibliotheken mit 400 gängigen TTL-Schaltsymbolen vereinfachen darüber hinaus das Zeichnen von Logik-Schaltplänen. Sämtliche Symbole können erweitert oder verändert werden, vollkommen neue Zeichen lassen sich jederzeit hinzufügen. Sehr hilfreich, wenn es schnell gehen soll ist die automatische Justierfunktion für das punktgenaue Ansetzen an vorhandenen Linien.
Das Software-Paket ist in drei Versionen erhältlich, für Rechner mit EGA-Karte (monochrome Darstellung), für Rechner mit CGA/Hercules-Karte und für Olivetti-PCs. Ansteuerbar sind alle marktüblichen Plotter mit bis zu acht Farben, als Behelf ist auch die Ausgabe auf einem IBM- oder Epson-kompatiblen Matrixdrucker möglich. Interessenten können sich für 20 DM eine Demo-Diskette kommen lassen.

Shamrock Software
Karlstr. 35
8000 München 2
☎ 0 89/5 11 73 77

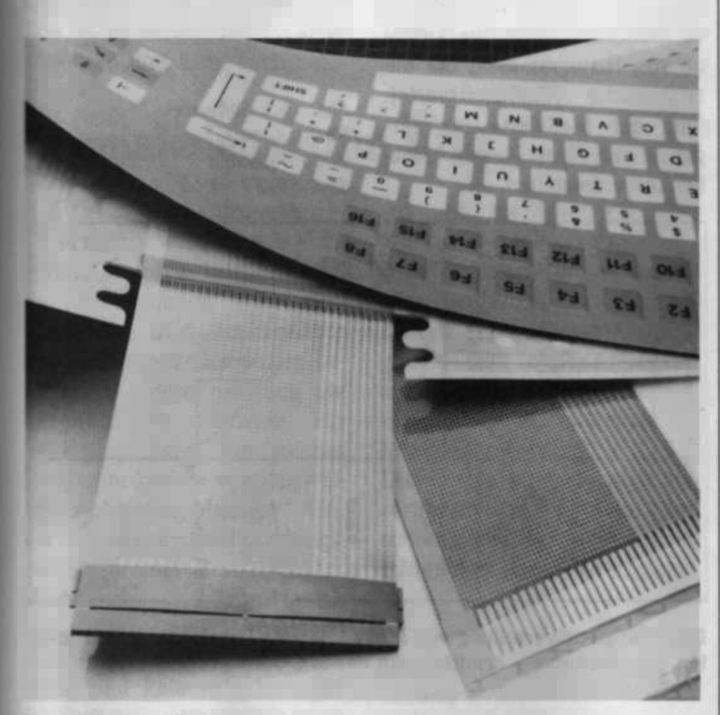


So sieht ein geplottetes Schaltbild des Shamrock-CAD-Programms aus

Transparente Abschirmfolie

Wo Sichtfenster, Computer-Bildschirme, Flüssigkristall- und Plasma-Anzeigen undurchlässig gegen Hochfrequenz-Einstrahlungen gemacht werden sollen, können neuartige Abschirmfolien Verwendung finden. Sie sind in verschiedenen – auch kundenspezifischen – Varianten erhältlich. Das Spektrum reicht dabei von einer im Siebdruck-Verfahren beschriftbaren Aluminium-Folie auf Polyester-Träger bis hin zu transparenten Metalloxyd-Beschichtungen. Eine deutliche Abschirmwirkung, so versichert der Hersteller, soll sich damit bis zu Frequenzen von 1 GHz ergeben.

R. H. Technical Industries
Easton Lane
Winchester, Hampshire
SO23 7RR, England
☎ 0 04 49 62/84 04 04



In vielfältigen Varianten gibt es bedruckbare oder transparente Abschirmfolien

Grafikprogramm für Atari ST

Ein leistungsfähiges Grafikprogramm für den Atari ST für etwa 100 DM gibt es jetzt vom Sybex-Verlag. Das Programm Starpainter ist für den monochromen Monitor ausgelegt und sehr einfach zu bedienen. Grafiken von anderen Programmen kann man für die Verarbeitung mit dem Starpainter umwandeln. Auch Farbgrafiken können so weiterbearbeitet werden. Selbst zeitaufwendige Funktionen wie z. B. das Füllen von Flächen oder das Vergrößern/Verkleinern von Ausschnitten führt das Programm sehr schnell aus. Bis zu acht DIN A4 große Grafiken kann man gleichzeitig im Speicher behalten und bearbeiten. Um Grafiken – auch ausschnittsweise – auf Papier zu bekommen, ist der Starpainter mit zahlreichen Druckertreibern und einer Druckeranpassung ausgestattet. Die Steuerung über ein jederzeit einblendbares Menü und das deutsche Handbuch erlauben, sofort mit dem Programm zurechtzukommen.

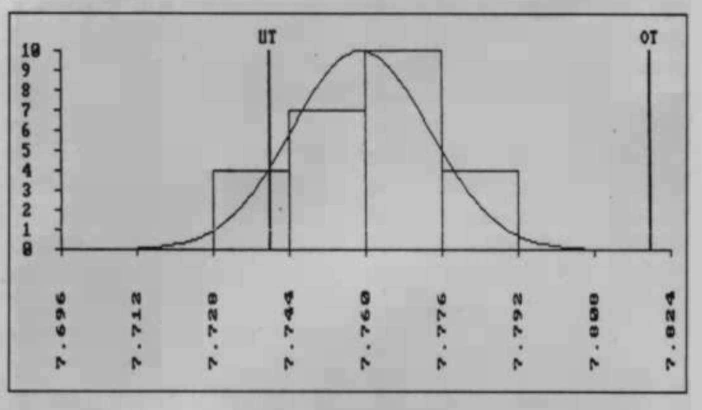
Sybex-Verlag
Postfach 30 09 61
4000 Düsseldorf 30
☎ 02 11/61 80 20

Qualitätssicherung mit Software

Auf allen MS-DOS-Computern lauffähig ist eine Reihe von Programmpaketen, mit denen hauptsächlich der Anwender in der Qualitätssicherung bzw. Fertigungstechnik angesprochen wird. Die Programme dienen der Führung von x-s-Regelkarten (Mittelwerte und Standard-Abweichungen von 25 Stichproben mit je 5...10 Meßwerten) für variable Merkmale, zur Kopplung mit np- bzw. c-Regelkarten für attributive Merkmale sowie zur Ausgabe von Histogrammen, die die tatsächliche Häufigkeitsverteilung der Meßwerte im Vergleich mit einer idealen Glokkenkurve darstellen.

Zur automatischen Meßwerterfassung dienen verschiedene Interfaces mit V.24-, BCD- und Mitutoyo-Schnittstellen. Bis zu zehn solcher Interfaces mit je 3...5 Meßgeräte-Anschlüssen lassen sich an einem einzigen seriellen Port des PC betreiben. Die hier genannten Programme stammen aus der SPC-Serie und befinden sich bereits bei einigen namhaften Firmen im Einsatz. Die Abkürzung SPC steht für Statistische Prozeßkontrolle.

Shamrock Software
Karlstr. 35
8000 München 2
☎ 0 89/59 54 68



Die Häufigkeitsverteilung von Meßwerten im Vergleich mit der idealen Glockenkurve – ermittelt von der SPC-Software

Disketten-Terminals zum Nachrüsten

Unter der Bezeichnung „Discmaster" kommen drei kompakte Disketten-Terminals auf den Markt, die über ihre serielle Schnittstelle an jeden PC anschließbar sind. Das erste



Über eine serielle Standard-Schnittstelle werden die Disketten-Terminals der Serie „Discmaster" an einen PC angeschlossen

Produkt nennt sich DS510, es beschreibt und liest 5¼-Zoll-Disketten, wobei die Bedienung manuell über Drucktasten (ähnlich wie die Aufnahme/Wiedergabe bei einem Kassettenrecorder) oder programmgesteuert erfolgen kann. Bei Tastenbedienung jedenfalls benötigt das Gerät keinen Softwaretreiber vom angeschlossenen Host-Rechner. Unabhängig vom Computer erfolgt in jedem Fall die Stromversorgung über ein eigenes 220-V-Netzteil.

Das zweite Modell nennt sich DS310, es verwendet 3½-Zoll-Disketten und kann auch netzunabhängig aus Nickel-Cadmium-Akkumulatoren versorgt werden. Hauptsächlicher Einsatzbereich dürfte deshalb die mobile Datenerfassung sein. Als Variante des DS310 präsentiert sich das Modell „Portadisc". Es arbeitet nur programmgesteuert, ist besonders kompakt aufgebaut und kann insbesondere im Zusammenspiel mit tragbaren Computern und Handterminals seine Fähigkeiten zeigen.

Neumüller
Eschenstr. 2
8028 Taufkirchen
☎ 0 89/61 20 80

Terminkalender im Griff

Wer Veranstaltungen – gleich welcher Art – zu organisieren hat, kennt den damit verbundenen Arbeitsaufwand und wird sich dankbar der Hilfe eines Computers bedienen – sofern natürlich die entsprechende Software zur Verfügung steht. Ein solches Programm könnte beispielsweise der „Meeting Manager" sein, der von den ersten Einladungsschreiben über Bestätigungen, Rechnungen und Zeitpläne bis hin zur Hotelreservierung und dem Ausdrucken von Themenlisten und Referatstexten alles im Griff hat – das umfaßt sogar das Drucken von Kuvert-Etiketten und die Übersicht über Währungs-Wechselkurse.

Dabei ist es egal, ob nur wenige oder mehrere tausend Teilnehmer zu erfassen sind. Das Programm läuft ohne Anpassung auf jedem Industriestandard-kompatiblen PC mit einer 10-MByte-Festplatte.

Die an sich schon recht benutzerfreundliche Bildschirmdarstellung wird durch ein gut verständliches Handbuch ergänzt.

Ambos
Radlsteg 1
8000 München 2
☎ 0 89/22 03 43



Mit dem Meeting Manager kann man Veranstaltungen jeder Art auf seiner 10-MByte-Festplatte organisieren

Beratung und Auftragsannahme: Tel.: 0 25 54/10 59 (Sammelnummer)

GESCHÄFTSZEITEN:
Montag bis Freitag von 9.00 – 13.00 Uhr und 14.30 – 18.00 Uhr. Samstags ist nur unser Ladengeschäft von 9.00 – 13.00 Uhr geöffnet (telefonisch sind wir an Samstagen nicht zu erreichen).

Sie erreichen uns über die Autobahn A1, Abfahrt Münster-Nord — B54 Richtung Steinfurt/Gronau — Abfahrt Altenberge/Laer — in Laer letzte Straße vor dem Ortsausgang links (Schild „Marienhospital") — neben der Post (ca. 10 Autominuten ab Münster/Autobahn A1).

Ein Preisvergleich lohnt sich!

Commodore

PREISSENKUNG: AMIGA 500 inkl. RGB-Farbmonitor PROFEX CM 14 S (Stereo, sonst techn. Daten wie COMMODORE 1081) nur 1595.–

AMIGA 2000, deutsche Tastatur, 1 MByte RAM, inkl. einem eingebauten Floppy 880 K, Maus, AMIGA-RGB-Farbmonitor 1084 und diverser Software nur **2795.–**

COMMODORE PC 40/AT, 1 MB RAM, dt. Tastatur, CPU 80286, IBM-AT-kompatibel, 1 Floppy 1,2 MB und 20-MB-Festplatte, inkl. 14"-Monochrom-Monitor, MS-DOS 3.2 und BASIC nur 3759.–
COMMODORE PC 1, 512 K RAM, dt. Tastatur, IBM-kompatibel, Farb- und Herculesgrafik, 1 Floppy 360 K inkl. MS-DOS 3.2 und BASIC 1148.–
Voraussichtlich in Kürze lieferbar:
NEU: COMMODORE PC 10-III, deutsche Tastatur, IBM-kompatibel, CPU 8088, 640 K RAM, zwei Floppys á 360 K 1789.–
COMMODORE PC 20-III, wie PC 10-III, jedoch 1 Floppy 360 K und 20 MByte Festplatte 2589.–
Während der Einführungsphase können beim PC 10-III u. PC 20-III Lieferzeiten auftreten!

TATUNG

TATUNG-Monitore auf Anfrage.

PLANTRON

PLANTRON-Computer weit unter den unverbindlich empfohlenen Verkaufspreisen von PLANTRON.

NEC

Die neuen NEC-Monitore auf Anfrage.

Schneider

SCHNEIDER-PC-1640-Serie, CPU 8086, IBM-kompatibel, 640 K RAM, deutsche Tastatur, Maus, komplett mit MS-DOS 3.2, GEM und diverser Software
MD/HD 20, mit einem Floppy 360 K, 20-MB-Festplatte und Monochrom-Monitor 2775.–
CD/HD 20, mit einem Floppy 360 K, 20-MB-Festplatte u. CGA-Farbmonitor 3198.–
ECD/HD 20, mit einem Floppy 360 K, 20-MB-Festplatte u. EGA-Farbmonitor 3775.–
Weitere PC-1640-Modelle auf Anfrage.
Voraussichtlich in Kürze lieferbar:
NEU: SCHNEIDER PC-2640-Serie, CPU 80286 (12 MHz Taktfrequenz), IBM-AT-kompatibel, 640 K RAM, deutsche Tastatur, Maus, komplett mit MS-DOS 3.3, GEM und diverser Software
mit einem 3½"-Floppy 1,44 MB, 32-MB-Festplatte und Monochrom-Monitor 4489.–
mit einem 3½"-Floppy 1,44 MB, 32-MB-Festplatte und EGA-Monitor 5289.–
Während der Einführungsphase können bei der SCHNEIDER-PC-2640-Serie Lieferzeiten auftreten.

ZENITH

NEU: ZENITH eaZy PC, 512 K RAM, CPU 8086-kompatibel (7,16 MHz), IBM-kompatibel, inkl. MS-DOS 3.2, GW-BASIC, MS-DOS-Manager, schwenkbarem Monochrom-Monitor
– mit zwei 3½"-Floppys á 720 K 1860.–
– mit einem 3½"-Floppy 720 K und 20-MB-Festplatte 2698.–

TAXAN

TAXAN-Produkte auf Anfrage.

SEAGATE

20-MB-Festplatte ST 225 inkl. OMTI-Controller 5520 nur 589.–
30-MB-Festplatte ST 238 inkl. OMTI-Controller 5527 nur 625.–
Weitere SEAGATE-Produkte auf Anfrage.

TANDON

TANDON-Business-Card

20-MByte-Harddisk-Steckkarte **648.–**
TANDON PCA 20, 1 MB RAM, CPU 80286, IBM-AT-kompatibel, 1 Floppy 1,2 MB inkl. 14"-Monochrom-Monitor, Monochrom-Grafikkarte, dt. Tastatur, MS-DOS 3.2, GW-BASIC und MS-Windows mit 20-MB-Platte 3875.–
Weitere TANDON-Produkte auf Anfrage.

ATARI

ATARI-ST/MEGA-ST-Serie weit unter den unverbindlich empfohlenen Verkaufspreisen von ATARI.

TOSHIBA

TOSHIBA-T1000-Portable, 512 K RAM, IBM-PC-kompatibel, Supertwist-LCD-Bildschirm (80 Zeichen × 25 Zeilen), ein eingebautes Floppy 720 K, Centronics- und RS-232-C-Schnittstelle, Akku-Betrieb 1998.–
Weitere TOSHIBA-Computer auf Anfrage.

VICTOR

Der neue VICKI:
512 K RAM, CPU 8088-2 (Taktfrequenz 4,77 MHz/7,16 MHz), mit 12"-Monochrom-Monitor, MS-DOS 3.2, BASIC
mit einem 5¼"-Floppy 360 K 1445.–
mit einem 5¼"-Floppy 360 K und 20-MB-Platte 2360.–

HANDY SCANNER

PREISSENKUNG: CAMERON-Handy-Scanner für IBM-kompatible Rechner, kompl. mit Interface, Treibersoftware und Scan-Programm nur 595.–

7 Monate Garantie auf alle Geräte!

EPSON

NEU: EPSON LQ 500 Matrix-Drucker 835.–

EPSON LX 800

Matrix-Drucker nur **535.–**
EPSON FX 800 Matrix-Drucker 925.–
EPSON FX 1000 Matrix-Drucker 1198.–
EPSON EX 800 Matrix-Drucker 1310.–
EPSON EX 1000 Matrix-Drucker 1640.–
EPSON LQ 850 Matrix-Drucker 1289.–
EPSON LQ 1050 Matrix-Drucker 1695.–
Weitere EPSON-Drucker auf Anfrage.

NEC

NEC-24-Nadel-Matrix-Drucker auf Anfrage.

olivetti

OLIVETTI DM 105 Farbmatrix-Drucker, 9 Nadeln, IBM- und EPSON JX 80-kompatibel, 120 Zeichen/Sekunde, anschließbar unter anderem an COMMODORE AMIGA 2000/500 nur 698.–

star

STAR NX 15 Matrix-Drucker 975.–
STAR ND 10 Matrix-Drucker 895.–
STAR ND 15 Matrix-Drucker 1195.–
STAR NR 10 Matrix-Drucker 1145.–
STAR NR 15 Matrix-Drucker 1395.–
PREISSENKUNG:
STAR NB 24-10 Matrix-Drucker nur 1198.–
STAR NB 24-15 Matrix-Drucker nur 1689.–
Auf alle STAR-Drucker gewähren wir 12 Monate Garantie.

OKIDATA

OKI-Microline-Serie und OKI-Laserdrucker in verschiedenen Versionen zu interessanten Preisen.

SEIKOSHA

SEIKOSHA SL-80 AI 24-Nadel-Matrix-Drucker nur 765.–
SEIKOSHA SL-80 VC für C 64 nur 765.–
Preise inkl. deutschem Handbuch.

C.ITOH

SUPER-RITEMAN F+III-Drucker inkl. deutschem Handbuch 695.–
Weitere C.ITOH-Drucker auf Anfrage.

BROTHER

BROTHER M 1409 Matrix-Drucker 789.–
BROTHER M 1509 Matrix-Drucker 945.–
BROTHER M 1709 Matrix-Drucker 1145.–
BROTHER HR 20 Typenraddrucker 989.–
NEU: BROTHER M 1724L 1365.–
Preise inkl. deutschem Handbuch.

QMS

QMS-Laserdrucker auf Anfrage.

JUKI

JUKI 6100 Typenraddrucker nur 745.–
PREISSENKUNG:
JUKI 5520 Farb-Matrix-Drucker 1095.–

CITIZEN

COMPUTER DRUCKER

PREISSENKUNG

CITIZEN LSP 100

Matrix-Drucker nur noch **570.–**
CITIZEN-Matrix-Drucker MSP 15e 845.–
CITIZEN-Matrix-Drucker 120 D 395.–
Preise inkl. deutschem Handbuch.
Neue CITIZEN-Drucker auf Anfrage.

Panasonic

NEU: PANASONIC KX-P 1540 24-Nadel-Matrix-Drucker nur 1445.–
Weitere PANASONIC-Drucker auf Anfrage.

TOSHIBA

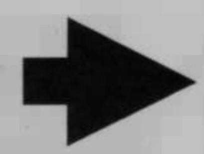
TOSHIBA-Drucker zu unseren bekannt günstigen Preisen.

FUJITSU

FUJITSU-Drucker auf Anfrage.

Fordern Sie bitte kostenlos die aktuelle Preisliste über unser gesamtes Lieferprogramm an oder besuchen Sie uns. **Selbstverständlich können Sie auch telefonisch bestellen.** Preise zuzüglich Versandselbstkosten. Versand per Nachnahme. Alle Preise beziehen sich auf den vollen Lieferumfang, wie vom Hersteller angeboten, soweit nicht ausdrücklich anders erwähnt.
Das Angebot ist freibleibend. Liefermöglichkeiten vorbehalten. Bei großer Nachfrage ist nicht immer jeder Artikel sofort lieferbar. Preise gültig ab 25. 1. 1988.

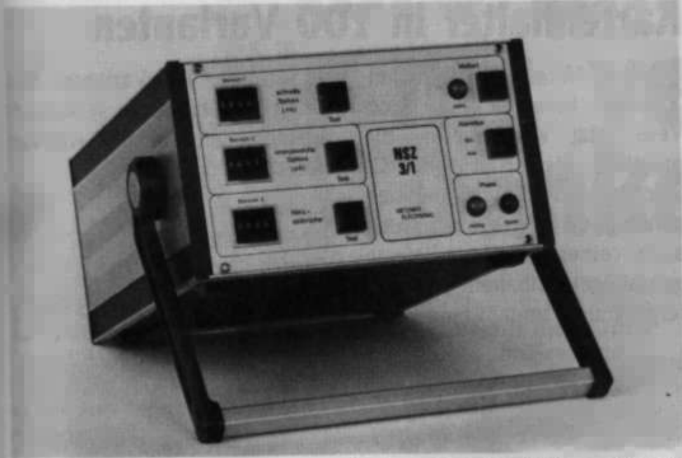
Benutzen Sie bitte zur INFO-Anforderung oder Bestellung die eingeheftete Antwortkarte.



MICROCOMPUTER-VERSAND

ernst mathes GmbH

Pohlstraße 28, 4419 Laer, Beratung und Auftragsannahme: Tel. 0 25 54/10 59



Recht preisgünstig kann man mit dem Netz-Stör-Zähler NSZ-3/1 Fehlern in der 220-V-Versorgung auf die Schliche kommen

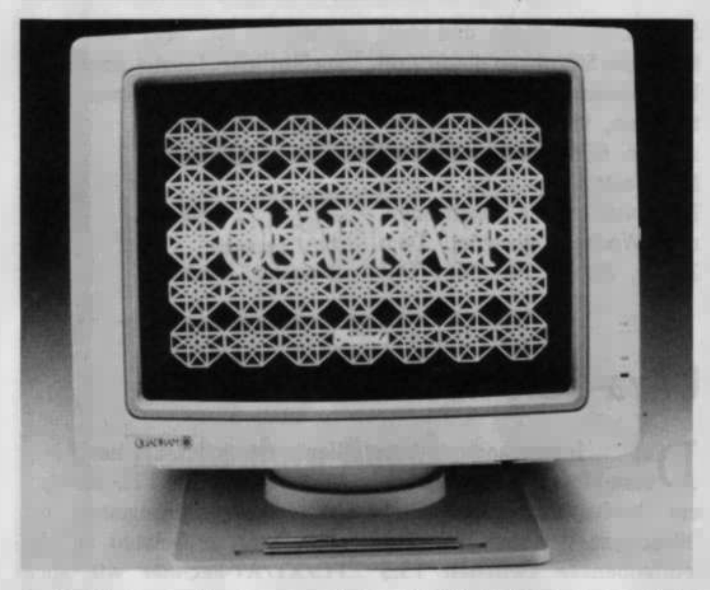
Störungs-Zähler am Stromnetz

Computerbenutzer, die oft mit unerklärlichen Hardware-Ausfällen oder Software-Abstürzen zu kämpfen haben, sollten vielleicht einmal Nachforschungen an der Netzsteckdose anstellen. Das geht z. B. mit dem Stör-Zähler NSZ-3/1, mit dem die drei am häufigsten vorkommenden Störkategorien erfaßbar sind: schnelle, energiearme Impulse im ns-Bereich, energiereiche Störimpulse im µs-Bereich sowie kurzzeitige Spannungseinbrüche und -ausfälle. Alle Störungen dieser drei Kategorien werden mit getrennten, vierstelligen Impulszählern fortlaufend registriert. Das Gerät ist auch für längere Kontrollzeiträume (sogar mehrere Wochen) geeignet. Dem Zähler, der für rund 800 DM erhältlich ist, liegt auch eine ausführliche Tabelle über mögliche Schutzmaßnahmen und -Geräte bei.

Metzner-Electronic
Postfach 11 14
7801 Umkirch
☎ 0 76 65/58 17

Ultra-EGA auch am PS/2

Der Interquadram-Multiscan-Monitor MS1420 ist ein hochauflösender 14-Zoll-Bildschirm, der sich allen Zeilenfrequenzen zwischen 15,5 kHz und 35 kHz automatisch anpaßt.
Seine vielfältigen Grafik-Darstellungsarten von 320 × 200 (CGA) über 640 × 480 (VGA) bis hin zu 800 × 600 Bildpunkten (Ultra EGA) stellen die Kompatibilität zu heutiger und auch zukünftiger Soft- und Hardware sicher. Da dieser Monitor sowohl eine TTL- als auch eine Analog-Eingangsstufe besitzt, kann er sowohl an alle PC/XT/AT-Rechner wie auch an Systeme der neuen Rechnerserie von IBM, IBM-PS/2, angeschlossen werden.
Mit dem neuen Quadram-Grafikadapter „Ultra VGA" läßt sich der für knapp unter 2000 DM erhältliche Bildschirm auch im VGA-Modus am XT/AT betreiben, was diesen Computern



In Verbindung mit einem speziellen Grafikadapter beherrscht der MS1420 auf herkömmlichen MS-DOS-Systemen auch den VGA-Modus

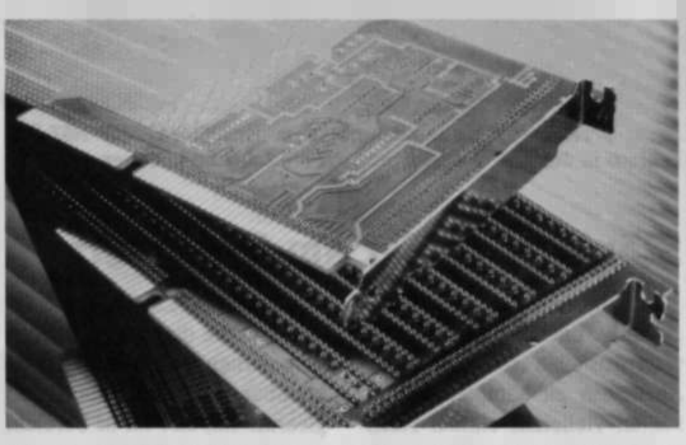
wiederum den Zugang zur VGA-Software eröffnet. Alle Bedienungselemente befinden sich in Griffnähe am Monitor. Die Bildfläche läßt sich in jede gewünschte Position bringen. Mit der entspiegelten Bildröhre läßt sich recht gut arbeiten.

Interquadram Computer
Hermannstr. 52
6078 Neu-Isenburg
☎ 0 61 02/1 70 95

Kartenhalter in 100 Varianten

Für XT/AT- oder PS/2-Computer kommen von Bicc-Vero jetzt Kartenhalter, die praktisch alle Arten von I/O-Steckverbindern aufnehmen können und je nach Ausführung auch einen Platinen-Befestigungsbügel enthalten. Das Lieferprogramm umfaßt eine Palette von über 100 Varianten aus poliertem Stahlblech – Sonderausführungen sind auf Anfrage lieferbar.

Bicc-Vero Electronics
Carsten-Dressler-Str. 10
2800 Bremen 61
☎ 04 21/8 40 70



Alle denkbaren Befestigungsvarianten lassen sich mit den PC-Kartenhaltern realisieren

Ihre Empfehlung für mc verdient ein Dankeschön

Der Erfolg von mc liegt in dem hohen Nutzen, den diese Zeitschrift professionellen Anwendern für ihre Arbeit bietet. Deshalb wird mc weiterempfohlen – an Freunde, Bekannte, Kollegen, die in irgendeiner Weise mit der Anwendung von Mikrocomputern befaßt sind oder daraus ein ernstzunehmendes Hobby gemacht haben. Dafür allen mc-Freunden ein herzliches Dankeschön.

Ab sofort möchten wir uns mit einem kleinen Geschenk bedanken, wenn durch Ihre Empfehlung jemand mc abonniert.

Sie haben die Auswahl unter den auf dieser Seite abgebildeten Dingen. Bitte kreuzen Sie gleich auf dem Empfehlungsscheck im Kartenbeihefter das von Ihnen gewünschte Geschenk an.



Rechner mit Bandmaß, schwarz, mit Schlüsselkette, Bandmaß 100 cm lang, Rechner mit 8stelliger LC-Anzeige, Speicher, inkl. Batterie.





Mini-Radio und -Rechner im Scheckkartenformat (FM-Card). Radio mit UKW-Empfang, Höhen- und Tiefenregler. Rechner mit Solarbetrieb, 8stellige LCD-Anzeige, 3-Tasten-Vollspeicher, Prozentautomatik.



mc-Sonderhefte „Turbo-Pascal". Ein Grundstock von Programmen für CP/M- und MS-DOS-Rechner. „Das Grafik-Sonderheft". Grundlagen, Tips, Anregungen und die neue Grafiknorm für den industriellen Einsatz.

Krypton-Taschenlampe, wasserdicht bis 30 m, sehr helle Krypton-Birne, mit Ersatzbirne und Umhängeschlaufe.

Diskettenbox mit 10 Disketten

Hinweis: Die Jahresabonnementgebühr für 12 Ausgaben mc beträgt DM 70,– (im Ausland DM 79,–). Eine Bestellung kann ich innerhalb von 10 Tagen beim Franzis-Verlag, Postfach 37 02 80, 8000 München 37, widerrufen. Zur Wahrung der Frist genügt rechtzeitiges Absenden des Widerrufs.



Alles, was ein Laptop-AT-Kompatibler so braucht, hat der 300 SLC drin

Neues aus der Laptop-Klasse

Von Rein Elektronik wurde unter der Typenbezeichnung 300 SLC ein AT-kompatibler Laptop vorgestellt, der für gut 8000 DM einen 10-MHz-80286 arbeiten läßt. Speicherseitig bereits ausgebaut sind 640 KByte (mit einer Erweiterungs-Option auf 2 MByte); als Massenspeicher installiert sind eine 1,2-MByte-Floppy (3½ Zoll) sowie eine 20-MByte-Harddisk. Angenehm das Erscheinungsbild des 640 × 400 Punkte umfassenden Plasma-Displays im EGA- und CGA-Modus. Und da auch ein Laptop hin und wieder Peripherie ansteuern muß, stehen zwei serielle sowie eine parallele Druckerschnittstelle nebst einem speziellen Interface für ein externes Diskettenlaufwerk zur Verfügung. Laptop für Anspruchsvolle.

Rein Elektronik
Lötscher Weg 66
4054 Nettetal 1
☎ 0 21 53/73 30

V.24 auf Stromschleife umgesetzt

Zu der DOS-kompatiblen, seriellen Schnittstellenkarte SSPC (mit ihr ist die softwaregesteuerte Umschaltung zwischen RS 232 C, Stromschleife und RS 422-Format möglich) gibt es jetzt einen mit Optokoppler-Potentialtrennung arbeitenden V.24/20-mA-Umsetzer. Dieser Zusatz wird in einem separaten Gehäuse zwischen dem 20-mA-Übertragungskabel und dem angeschlossenen Gerät eingesetzt. Seine Optokoppler sorgen dafür, daß äußere Einwirkungen auf die Übertragungsleitung (z. B. indirekte Blitzschläge) nicht mehr an die restliche Systemhardware weitergegeben werden. So lassen sich häufige und oftmals unerklärliche Störungen oder Schäden an Schnittstellen, Druckern oder Terminals vermeiden.
Die nach beiden Seiten wirksame galvanische Trennung kann aber auch ohne die Schnittstellenkarte SSPC erzielt werden,



Potentialtrennung mit Optokopplern im V.24/20-mA-Umsetzer gibt zusätzliche Sicherheit für die Schnittstellenkarte SSPC

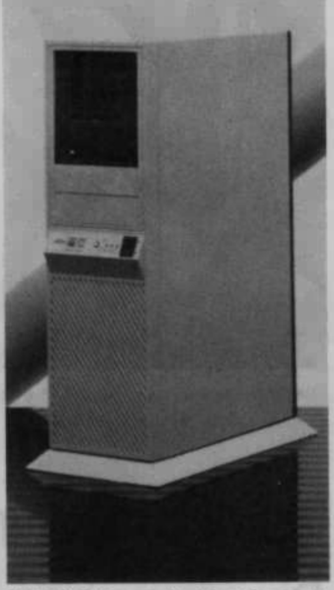
indem man am Anfang und Ende der gefährdeten Übertragungsleitung je einen Umsetzer vorsieht. Dies hat zudem den Vorteil, daß die Potentialtrennung auch dann erhalten bleibt, wenn der angeschlossene Rechner ausgewechselt werden soll-te. Der Preis für einen Umsetzer liegt bei 220 DM, die Schnittstellenkarte SSPC ist mit 395 DM zu veranschlagen.

IDE GmbH
Isselhorster Str. 403
4830 Gütersloh
☎ 0 52 41/6 86 09

80386 im Tower-Gehäuse

Premiere erlebte unlängst der Peacock 80386, ein im Tower-Standgehäuse untergebrachter 16-MHz-Computer mit sechs halbhohen Peripheriegeräte-Einschüben an der Frontseite. Der Peacock 80386 basiert auf einem Original-Intel-Motherboard mit Phoenix-ROM-BIOS, 512 KByte RAM, Echtzeituhr, Seriell- und Parallel-Schnittstelle sowie mit einem Anschluß für AT-kompatible Tastaturen.

Der fast schon obligatorische Sockel für den mathematischen Coprozessor 80387 oder ein SBC-Math-Modul ist natürlich vorhanden.
An Erweiterungsplätzen stehen dem Anwender zwei 8-Bit-, vier 16-Bit- und zwei 32-Bit-Slots zur Verfügung.



Mit 16 MHz arbeitet der Peacock 80386 im Tower-Gehäuse an seinen 32-Bit-Daten

Audio-Impex
Friedrich-List-Str. 67
4790 Paderborn
☎ 0 52 51/5 00 50



Die Softwarepakete Imagelab und Imagetool bieten viele komfortable Bildverarbeitungs-Funktionen

Bildverarbeitung mit Komfort

Bei Stemmer sind jetzt die beiden Softwarepakete „Imagelab" und „Imagetool" zur Bildverarbeitung erhältlich. Das letztgenannte enthält eine Bibliothek von über 100 Verarbeitungs- und Manipulations-Funktionen, die ihrerseits in Imagelab benutzt werden. Damit kann man Mittelungen, Subtraktionen, die Bewegungskontrolle, Kontrastverbesserungen, Pseudofarben-Darstellungen usw. durchführen.

Eine eigene Datenbank hält sämtliche Bildsequenzen für das System bereit, der Lernmodus registriert ablaufende Operationsfolgen und übersetzt diese selbständig in Makro-Funktionen. Alle Imagetool-Routinen sind von anderen Programmen aus aufrufbar, die in Microsoft Fortran 77, Pascal oder C geschrieben sind, was zeitaufwendige Programmierarbeit verhindert.

Stemmer PC-Systeme
Boschstr. 12
8039 Puchheim
☎ 0 89/80 90 20

Per Optokoppler in den PC

Zum Preis von 368 DM ist bei Kolter Electronic eine Optokoppler-Zusatzkarte für PC/AT-Systeme einschließlich Demo-Software erhältlich. Mit den 24 Koppler-Bausteinen sind maximal 16 Eingänge sowie acht Ausgänge unter Kontrolle des Port-Bausteins SAB-8255 gleichzeitig programmierbar. Da die auf der Platine untergebrachten Widerstands-Netzwerke nach Belieben getauscht werden können, läßt sich die Karte an verschiedene Spannungs- oder Strombereiche der Eingangssignale anpassen. Bei Auslieferung befinden sich 100-Ω-Netzwerke auf der Platine – das ist die Standard-Konfiguration für 5-V-TTL-Eingänge. Über Steckbrücken kann der Anwender die Karten-Adresse frei programmieren, so daß letztlich auch mehrere solcher Optokoppler-Platinen in einem PC installierbar sind. Im Grundzustand ist die Standard-Adresse 0300 H vorgegeben.

Kolter Elektronik
Postfach 11 27
5042 Erftstadt
☎ 0 22 35/7 67 01

CMOS-RAM mit Batteriepuffer

Speziell für den Einsatz in industriellen Steuerungssystemen auf der Basis eines IBM-PC (oder eines Kompatiblen) hat HSH-Systeme eine statische CMOS-Speicherkarte entwikkelt. Der (über die PC-interne Stromversorgung) gepufferte Akkumulator sichert dabei den Datenbestand auch nach dem Abschalten des Rechners. Ihre maximale Speicherkapazität beträgt 32 KByte und ist in 8-KByte-Schritten über den gesamten 1-MByte-Bereich adressierbar.
Als Zugriffszeit spezifiziert das Datenblatt 120 ns, durch die variablen Wartezyklen kann auch eine Anpassung an unterschiedliche Rechnergeschwindigkeiten erfolgen.
Dem Datenschutz beim Rechnerstart dient bei dieser Baugruppe übrigens ein „I/O-Schloß", das erst nach einmaliger Adressierung den Zugriff auf das RAM freigibt.
Die Karte läßt sich überall dort vorteilhaft einsetzen, wo nach einem Spannungsausfall weder Programmteile noch Daten zerstört werden dürfen, aber andererseits aus Zeitgründen die Sicherung auf einem externen Massenspeicher nicht erfolgen kann oder soll.
Vor allem für kleine Systeme sehr interessant.

hsh-Systeme
Bismarckstr. 43
7100 Heilbronn 1
☎ 0 71 31/10 00 34

mc-MINIMARKT

SUCHE SOFTWARE

Suche NC-EDITOR bzw. CNC-PROGR. zum Programmieren von CNC-Dreh-, -Fräs- und -Bearbeitungsz. für FANUC 10 Steuerung mit PC-XT.
☎ 02 11/23 27 23–4 30 75 24

BIETE AN SOFTWARE

■PC-Freesoft Disk 2.80 – 4 DM■ ■150 W Netzgerät 129 DM/zus.:■ ■XT Klappgeh. 129 DM/248 DM■ ■BabyAT-Geh., 200 W, 339 DM■ ■ATboard+Phönix ab 648 DM. Liste frei! G. L., 8501 Veitsbronn, Bachleite 16 (G)

400 C64-Computerspiele und Anwenderprogr. auf 25 Disketten nur 490 DM. ☎ 0 51 82/33 77.

Dongle-Keykarte defekt? Wir helfen! Personal Designer 350 DM, Novell 280 DM, CadVance 190 DM, P-Cad 320 DM, EEDesigner 190 DM, Protel PCB Editor 145 DM, Robocad&Solid 215 DM, Redboard 235 DM, Cadkey +Cadstar 190 DM, Schemall 185 DM. Autocad 165 DM, PC2 225 DM, PLC 2 Ladder 245 DM, Eplan 350 DM, PC-Draft 250 DM. Info anford.! * Bitte Urheberrechte beachten * **!Deutsche Programmanleitungen!** 1Dir V3.5 45 DM, VFeature de Luxe V2.5 45 DM, Copyll Copy Board V4.3 35 DM, Fastback 30 DM, Sideways V3.1 30 DM, Norton Commander V1.02 30 DM, Norton Advanced Utilitie V4.0 40 DM, PCTools V3.24 35 DM, Copywrite 11/87 25 DM, Copyll PC V4.01 25 DM, Flugsimulator 40 DM, Newsroom 30 DM, Disk Optimizer 25 DM, Paintbrush 35 DM. Versand NN/Scheck + 5 DM. Ing.-Büro Stock, Postf. 10, 7926 Böhmenkirch, ☎ 0 73 32/50 78, Btx *073 325 079# (G)

Lehrer! Noten berechnen, verwalten, listen mit „SCHULDAT". Kl.-Liste, Geb.-Liste, Adressen. Menüsteuerg. auch f. Laien, f. PCs nur 39.– DM. Info 80 Pf. D. Arnold, A.-Dürer-Str. 38c, 7835 Teningen 1 (G)

PASSWORT-SCHUTZ 100% für MSDOS, kein Starten ohne Passwort, 149.– DM. Info: R. Münk, Ing.-Büro, Brunnostr., 8702 Margetshöchheim, ☎ 09 31/46 24 77, Händleranf. erw. (G)

miroECBplus: Master, DRAM, ROM/RAM, FDC/SASI, GDC, Videoplus, 2× LW-Teac-135F, Terminal-Ampex 230, Preis: VB. ☎ 0 53 68/2 71, ab 18 Uhr

PC 1600 KOPPLUNG MIT IBM XT/AT, LOAD/SAVE von Programmen, TEACH über PC, LPRINT, LLIST auf PC-Drucker mit Anleitung 49.– DM. Gerhard Rieder, Hauptstr. 3, 5441 Ueß, ☎ 0 26 92/3 12

Apple 2+ komp. (Basis 108) mit 10 MHz Z80-Karte, 6809-Karte, 128k-Saturn 2× Teac 55F auch einzeln. VB 1700 DM. ☎ 02 08/7 18 41, ab 18 Uhr außer Di.

Professionelles Leiterplatten-Entflechtungs-Programm für AT (XT) von Racal-Redac (NP > DM 10 000.–) günstig um 2500.– abzugeben. Ing. Müllner, Mozartstr. 6, A-3385 Prinzersdorf

Cross-Assembler f. MSC48, MCS51, Intel-komp. ab 139 DM, CP/M, MSDOS. Info: H. Schröder, K.-Jäger-Str. 14, 4790 Paderborn, ☎ 0 52 51/7 28 88 (G)

GEM-Shell (Oberfläche) für Assembler-Programmierer auf ATARI ST. Info gegen Freiumschlag bei Tilo Baumann, Hardenburgweg 4, 6729 Jockgrim

ATARI ST: GfA Basic Compiler (neu, Ver.) DM 60, Modula-2 Dev. V.3.0 DM 250, Lattice C V.3.04 DM 150. R. Scholz, ☎ 0 82 21/64 92

CP/M- und MS-DOS-Software: FIBU/Datenverw./Grafik/und ... Liste anfordern! ☎ 0 42 35/83 14 (G)

IBM-komp. Lernprg. Techn. u. Graf. zu reel. Preis-Vokab. Mini-CAD, Mathe, Physik, Chemie-Kat. 1 DM. A. Ristau, Peetzweg 9, 3320 Salzg. 1 (G)

Wir brauchen Platz! Räumungsverkauf orig. u. gebr. Software (dBase, Editoren, Assembler, Compiler u. v. a. m.). Liste anfordern! PASCAL Consulting, ☎ (0 89) 1 29 30 78 (G)

Publ. Dom. 4.–/St. + NN Auflös., Privatbest., Karte m. Titel + Tel. Panzer, Kobolastr. 21, 8900 Augsburg

1300 Public-Domain-Disk f. PCs, 6.–/Disk. Katalog (5 Disk) 10.–. Info 1.– i. Brfm. Lindmeier H., Fichtenstr., 8312 Dingolfing (G)

FORMATKONVERTER: Schreiben und Lesen v. **CPM**-Disk. auf **IBM-PC/XT/AT.** Formate frei definier- u. abspeicherbar. **DM 128.–,** Info bei: **HüWiCo,** Am Siepen 17, 4630 Bochum, ☎ 02 34/36 12 06 (G)

NEU dbTOPSCRIPT NEU Einzel-/Serienbriefe in dBASE3/dBASE3+ mit Adreß-Selektion, Briefkopfgestaltung, Anrede usw. mit Suchroutine, Etikettendruck, Menüführung, nur auf 5¼-Disk für IBM PC. DM 50 (Scheck) Vorauskasse. Ehrig, Neuwiese 27, 8918 Riederau

Unglaublich, aber wahr! Über 1000 Disketten mit PC-Freiprogrammen ab 5 DM lieferbar. Der neue Katalog ist schon da, kostenlos, wie's immer war. Telefon rund um die Uhr 0 52 61/8 89 01, wagen Sie's nur. Günther, Braker Mitte 28C, 4920 Lemgo (G)

➢ **MC-EPROMMER (1.87) unter OS9/6809** ≪ Menü BASIC09, Assemblertreiber für PIA, Source: DM 35.– (3,5") bzw. 32.– (5,25"). Info ☎ 0 72 43/44 27 ab 18 Uhr

Haftetiketten, 40 Größen, Info: Böhne, Frösterweg 4, 3354 Dassel (G)

Z80-fig-FORTH (CP/M), frei geg. form. 8"-/5¼"-Disk u. Rückporto. E. Ramm, Postf. 38, 2358 Kaltenkirchen, ☎ 0 41 91/16 21

PUBLIC-DOMAIN-SOFTWARE für IBM-Katalogdisketten gegen 5.– DM. Funkcenter Mitte GmbH, Klosterstr. 130, 4000 Düsseldorf (G)

Fax auf dem Atari-ST. SW-Bilder senden und empfangen bis DIN A2, Ausdruck auf jeden Drucker. HAMFAX-ST 35.– DM. B. Schmidt, DD4DZ, Bergstr. 40, 4690 Herne 1, ☎ 0 23 23/4 34 58

SUCHE HARDWARE

Suche **PL/1-86** Dokumentation, auch leihweise bei Kostenübernahme, ☎ 0 40/7 45 75 15

TEAC-Laufw. FD 55, 360 KB, für PC gesucht, ☎ 0 22 03/2 38 73

BIETE AN HARDWARE

Apple-komp., Z80, 80Z, Centr., 2 LW, abgesetzte Tast. z. vk., 850.– DM, NDR-CP/M-fähig als 68008 oder Z80-System, Preis VHS. ☎ 0 41 52/7 91 75 ab 19 Uhr

HEIZUNGSREGLER, selbstlernende Hausheizungsregelung mit Absenk- und Warmwasserfunktion. Der Regler ist als Bausatz oder als fertiges Gerät erhältlich. Hacker electronic, Am Eichholz 6 8653 Mainleus, ☎ 0 92 29/10 00 (G)

NDR-Computer: DYNRAM 256 = 100.– DM, Bus 3 = 100.–, HCOPY = 120.–, NETZ 2 = 120.–, BANKBOOT = 35.–, CAS = 30.–, STAR SG10 = 320.–, ATARIMAUS = 90.–. LITERATUR:
☎ 05 31/33 72 14

mc-MINIMARKT

PC-XT 10 MHz, 640 KByte, 1 FDD, Hercules P.-Card 14"-Monitor amber, **1549 DM,** 14"-TTL-Monitor **299 DM,** Preisliste anfordern. ☎ **0 86 62/87 47**, ab 17 Uhr [G]

1 Apple-Drucker Scribe für Apple 2c, neu 250.– DM. ☎ 0 20 52/ 16 50

ITT 3030, CP/M 2.2, Turbo Pascal, VB 1400.– DM. ☎ 0 60 31/6 25 41

CPM/3-Komplettsystem 2× 1 MByte Floppy, Monitor, IBM-Gehäuse, SNT, eig. Terminal, Graphik 640×400, prog. Tastatur, div. Software, VB 1850 DM. ☎ 0 52 61/1 54 42, ab 18 Uhr

MC 68000, voll funktionsfähig, 512 KByte, 2 LW, Tast., Mon., viel SW, VB 1699.– DM. ☎ Silvio, 0 89/32 60 12

MC68000, 512 KByte, 2 LW, 1 MB, HARDDISK 66 MByte netto, Epromkarte, Sprache-Uhrenkarte, Busterminierung, CP/M 68k, BIOS 2.0, Utility-Disks, + viel Software, 1A-Zustand, Neup. 10 000 DM, VB 5200 DM. ☎ 0 89/3 54 25 68

EGA-MONITORE , DM 850.–. ☎ 0 70 31/5 16 56

Commodore 610, 256 KByte, + SFD 1001 + Kabel 750.– DM + Drucker 3022 150.– DM. Steinmeier, ☎ 05 41/4 54 96

CAD-Ö+R-CPM-Rechner mit EPC/RGB, Tastatur, Bildschirm, 3½"- + 5¼"-Floppy, Beschreibung und Software kompl. 1200.– DM. ☎ 0 61 61/28 53, ab 18 Uhr

MC68000, 512 KByte, FDC, Cherry-Tast., CP/M + Z80 Emu., Preis VB; Z80-Sys. kompl. 2×80 Tr. HW-Unkosten 399.– DM; IBM-XT kompl., 640 K, 20 MByte, FARBMON., div. Schnittst., SW, V20-PROZESSOR + BIOS, Preis: 2950.– DM VB. ☎ 0 60 21/ 5 73 34, nach 18 Uhr

Gewerbl. Anzeigen sind mit [G] gekennzeichnet.

MC68000, Term., Spr. & Uhr, 512 KByte, 2 LW, NT3, PASCAL, BASIC, FORTRAN, FORTH usw. + viel Z80-Softw., nur komplett, VB. ☎ 09 31/28 57 87

MC68000-Rechner 0,5 MByte, RAM 2×1 MByte, Floppy, Term., Mon., Geh., Tast. NT2, viel SW CP/M 68k, DM 1699 (auch einzeln). ☎ 0 21 06/7 18 69

ADI-DM 14 MONITOR und HERCULES-Karte DM 350.– für Selbstabh. ☎ München 0 89/9 57 09 46

MC-CPM (SYS, OUT, FLO) + 5,25"-LW, 370 DM. ☎ 0 75 31/3 28 91

*** HEIZUNGSREGLER ***
* selbstlernende Haus- *
* heizungsregelung mit *
* Absenk- und Warmwasser- *
* funktion. *
* Der Regler ist als Bausatz *
* oder als fertiges Gerät *
* erhältlich. *
* *
* Hacker electronic *
* Am Eichholz 6, 8653 Mainleus *
* ☎ 09229/1000 [G] *
****

NDR 68008, 1 LW, Prommer, 22-MHz-Monitor, Cent, gr. Keyboard, kpl. m. Unterl. DM 800.–. ☎ 09 11/30 54 37

* CBM-SYSTEM-ANWENDER * Softbox f. CP/M-Betr.-System u. V24 incl. Disketten f. CBM8050, Handbücher und Microsoft-Basic u. Assembler, VB 800.– DM. ☎ 0 22 43/1 24 75

SCHALTNETZTEILE, 5 V, 20 A; ±12 V je 1 A; 35 V, 3 A, nur 69 DM. ☎ 0 51 46/86 81 [G]

ORCHID-Turbo-EGA-KARTE für PC/XT (80286), DM 800. ☎ 0 30/ 8 01 57 13

***** TURBO-KIT! ***** f. IBM-kompatible XT (Norton 3.0) 79.– DM. ☎ 0 60 74/7 02 09

WANG 2200T incl. 1 Terminal u. Fest-/Wechselplatte, Typ 2260B, VB 1200.– DM. ☎ 0 21 01/1 61 17

HP-Plotter, 7475 A, neu mit Garantie, DIN A3/A4, 6farbig, 4898.– DM. ☎ 0 23 04/1 76 55

Apple II + Orig., 2 Floppy, 2×40 Tr. Ehring, Ser, Par, Z80, 80Z, Prommer, Monitor 12" 60 D, VB 1200.–. ☎ 0 41 02/6 44 81

MC68000 kompl. mit Monitor, Drucker und Tastatur, CPU 68000, 512 KByte, 2×Floppy 6138, CP/M, C u. a., VB 3700.– DM. ☎ 07 81/ 3 12 64 (07 81/8 32 20 gesch.)

CBM 1001 Single Floppy (1 MB) DM 700.–, CBM 8296 DM 1100.–, CBM 8028 Drucker DM 600.–, alles neuwertig, auch einzeln. Maurer, Berlin 30, ☎ 0 30/2 13 79 00

APPLE-IIe-System, 128 KByte, 2 Drives, alles Orig., Mon., Z80 + viel SW, für 1950 DM. ☎ 0 73 62/41 01

EPSON HX-20
Preis VB
Ulrich Domröse
Sandstr. 108
6102 Pfungstadt
☎ 0 61 57/31 55

CBM 8032 + DOPPELFLOPPY 8050, VB 1200 DM. ☎ 0 20 45/79 69, nach 18 Uhr [G]

Verkauf aus Lagerüberbeständen: Floppy-, Festplatten-Laufwerke, Monitore, Tastaturen usw. zu sehr günstigen Preisen. Bitte Liste anfordern. Uhlenbrock Elektronik. ☎ 0 20 45/70 91 [G]

mc-MINIMARKT

BIETE AN HARDWARE

APPLE II Space 84, 192 KByte IBM-Geh., progr. Tastatur, Ehring-Contr., 1 TEAC-LW 80 Spur. 80 Z., Z80H, PAL u. Lit.
☎ 05 21/20 01 86

Verkaufe 2-MByte-RAM-Karte für MC68000, aufgebaut, nicht getestet, für 300.–, MC68000-Board, 128 KByte, für 400.–.
☎ 06 81/89 31 73, ab 17.30 Uhr

Original-IBM-Portable mit 20 MByte, 2×720 KByte, 3,5"-Floppy, 640 KByte, V20, Uhr, Game, V24, Printeradapter, CGA-Karte, evtl. ATI-Wonder, neuwertig, VB 3900 DM. ☎ 02 21/4 89 05 26 oder 0 22 73/22 29 (n. 19 Uhr)

MC68010, 512 KByte, Uhr, 2× 5,25"-LW, Geh., Tast., Mon., A-Joystick, Softw.: OS9, CP/M 68k, Forth usw., VB 2200 DM.
☎ 0 89/52 88 35

NDR CP/M System m. Z80H, 8 MHz CPU 64 KByte RAM, 240 KByte RAM, Flop., 2 LW 3,5", EPSON 2×86 PC-Gehäuse, SW: WS 3.0 dBASE II, u. a. viel Literatur, DM 3500.–. ☎ 0 52 25/17 69

Apple II+ Komp., PC-Geh., ext. Tast., 2 Teac-LW 55 FV, FDC4-Contr., Z80, 80Z, Par.-Karte, 150-W-Netzteil, mit 12"-Monitor, DM 1000.–, verk. auch einzeln, 10 MByte HARDDISK m. F+B-Contr. DM 500. ☎ 0 96 04/18 80

NDR-Computer CP/M+, Vollausbau, Europac-Geh., Teac-Floppys, große Tastatur, Software 2000 DM, V24-Drucker 150 DM, SW-Monitor 100 DM.
☎ 0 61 47/27 36, ab 19 Uhr

Apple-IIe-Komp. mit Monitor u. externer Tastatur, 64 KByte, Z80, 80 Z., serielle Schnittstelle, 2 Laufw. à 640 KByte, mit viel Literatur, VB 1500 DM.
☎ 02 21/48 19 04 (ab 18 Uhr)

CP/M PLUS O/R, 128 KByte, 6 MHz, Uhr, 2×1, 2 MByte Shugart, 5"-Laufw., m. Software, verkaufe oder tausche gegen Drucker.
☎ 0 04 54/68 56 97

EINMALIGE GELEGENHEIT!
Alphanumerisches Plasma-Display, 8 Zeilen × 40 Zeichen, ASCII. Industrieausführung betriebsbereit, VB 180.– DM. Anfragen unter ☎ 0 28 61/41 91

NDR-Klein-Computer CP/M-Vollausbau, Z-80-CPU, GDP 64 KByte, ROA 64 KByte, CAS, Bank-Boot, 2×10E, FLO-2, Hcoppy-Maus, SER, 2×RAM 256 KByte, SBC-3, Prommer, 1200 DM.
☎ 0 81 61/41 06, ab 18 Uhr

LW Siemens FDD 100-8 SS/DD, 500 KByte, VHB 350 DM. ☎ 0 43 42/53 44

---

2 Orig.-IBM-Disk-Lfw. 360 KByte à 250.– DM zu verkaufen, evtl. mit Orig.-Contr. 100.– DM.
☎ 09 11/67 37 25

SHARP PC7000.
☎ 05 61/31 29 69

NDR-Computerteile zu verkaufen, voll funktionsfähig! POW5Y, CPU Z80, BUSII, IOE/EX, CAS, Bank-boot DRAM 128/Software! PASCAL/S, DIS6 B/SOGI-DOS, Basic, Gosl, MONI.
☎ 07 31/6 36 77, ab 19 Uhr

Gewerbl. Anzeigen sind mit [G] gekennzeichnet.

Apple II+: 128 KByte RAM, Erphi-Ctrl. mit SW + Lit., je 80 DM.
☎ 0 29 47/35 79

Apple II+, 64 KByte, 2 LW, viel Hard- u. Softw.
☎ 0 69/89 73 71, abends

SIEMENS PCD, 1 MByte RAM, 13 MByte HD, Maus und viel Software, DM 5000. ☎ 02 21/49 14 15

****** C-64 *****
EPROM-SIMULATOR f. 2716 bis 27256 incl. Softw.: 6502/10/C02/01Q/AssMon/Z80, Gratisinfo: Dipl.-Ing. J. Kamnig, Postfach 4, A-8019 Graz, Österreich
****** C-64 ******

NDR-Computer zu verkaufen, auch Teile, SBC2, CPU 68 KByte, ROA64, KEY, TAST1, GDP 64 KByte, Prommer, CAS, NEZ, GEH, 2×BUS2, IOE, Lit. ☎ 0 40/7 96 38 62, nach 17 Uhr

---

CP/M-MZ80K-System, 48 KByte, 2×5¼"-LW, Drucker, DM 800.
☎ 0 91 31/72 25 13

EPROM-SIMULATOR f. 2716 bis 27512 (o. Gehäuse) **248.– DM, 512K-CMOS-RAM f. ECB-BUS,** direkt adressierbar, fertige Platine, ohne RAMs **298.– DM.** Info bei: **HüWiCo,** Am Siepen 17, 4630 Bochum, ☎ 02 34/36 12 06 [G]

Seikosha Graphic Printer GP250x, ca. 1 Jahr, Schnittst., Centronics, RS-232C, 250.– DM.
☎ 0 44 31/25 94, ab 18 Uhr

MC-CP/M-Computer wegen Hobby-Aufgabe zu verkaufen, bestehend aus SYS1, OLT1, TERM1, FLO1 + NT im 19"-Einschub, Preis VHS. ☎ 0 56 45/5 21, ab 18 Uhr

---

SCHWEIZ: MC68010 Multilayer, 1,5 MByte, 3 LW, Z80-Karte, Uhr, CP/M 68k, div. Hard- u. Software, Preis VB. ☎ P 01/3 02 25 04, G 01/2 45 37 64, Hr. Buckmann

---

CBM 8296-D DM 1600.–
CBM 8096 + 8250 DM 1000.–
Brill, ☎ 0 24 02/8 10 14

Conitec PROF-180X, 9 MHz, 650 DM, **GRIP-4** 350 DM.
☎ 0 82 71/57 76

mc-MINIMARKT

NICOLET Logic Analyzer, 200 MHz, 48-Kanal-State, 16-Kanal-Timing mit allen Optionen, NP 80 000 DM, Gerät ist neu, Angebote nicht unter 30 000 DM. R. Brüggemann, Kard.-v.-Galenstr. 20, 4450 Lingen

HX-20, 32K, MC, Drucker, Programme, kompl., techn. Handbuch, VB. MC68000, FD-Contr., Bus-Term., Monitor, Tast., 1 FD 80 TR, NT mit CP/M 68k, OS-9, Programme, VB. ☎ 0 61 03/2 22 19, nach 18.30 Uhr

MINI-286-Board, 6/8/10 MHz, 1 MByte bestückt, BIOS mit integr. Set UD und deutsch. Zeichensatz, VB 1250 DM. Postkarte mit Tel.-Nr. an Waldemar Affa, Von-Der-Recke 40/301, ☎ 0 21 03/12 80 13, Zi. 301

Vierfarbplotter NPR 5500, DIN A4, mit serieller und paralleler Schnittstelle für DM 800.– zu verkaufen. ☎ 0 75 31/4 31 75

CP/M PLUS 3,0 Karte für Apple II+ oder Komp. mit div. Dienstleist.-Progr. DM 380.–. ☎ 0 41 54/29 78

CP/M 3.0/2.2 Wavemate-Bullet, Term1, Monitor, 2× Teac 55 F., IBM-Tast. in IBM-XT-Geh., incl. Man., div. Software. ☎ 0 21 73/2 48 77

NDR-Computer, 4× RAM 256 dyn. je 225.– DM. Ahrens, ☎ 0 40/58 82 85

Orig.-Apple II+, 64 KByte, 2 Laufw., Haga-Monitor, Epson-Drucker, viel Softw. (Faktu., Text, Kalk., Datenb., Spiele), VB 1990.–. ☎ 0 89/6 11 53 00 p, 50 33 93 d

MC68010, 512 KByte, **20-MByte-Harddisk,** , Busterm., AD/DA-Wandler, 1×FD, Uhr, 8A-Nt., schönes Gehäuse, Monitor, VB 3400.–; ebenfalls abzugeben: 2 MByte RAM, 8fach-V.24 (auch einzeln). ☎ 09 11/68 50 72

Apple II+, 64 KByte, ext. Tast., 2 LW, Z80, 80 Z, Drucker-Int. Epromer, Softw., Literatur, VB 1790.–. ☎ 09 11/51 29 12, ab 17 Uhr

MC68010-Motherboard, Floppy-Controller, SASI-Interface, Terminierung, NT 2, Gehäuse und Dokumentation, funktionsgeprüft durch DSM, 800.– DM. ☎ 05 51/3 51 31

Magnetkartenleser. ☎ 05 71/4 98 56 Ⓖ

IBM-Kompatible komplett: DER AT **80386 komplett 8700.–,** Mini AT 2100.–, XT 1600.–, 30 MByte 740.–, NEC-Multisync 1380.–, Händleranfragen erwünscht. ☎ 0 63 03/43 87 Ⓖ

MC-CP/M, LF40, OUT, SYS-6 MHz), 10A-Nt., 8"-FD, prof. 19"-Geh., Terminal incl. Tast., Monitor, viel SW, VB 600.–. ☎ 09 11/68 50 72

Externer 1-MByte-Printerbuffer
- serielle Schnittstellen
- alle Protokolle 790.– DM

☎ 05 31/37 70 07-09

286-Speed-Karte nur 798.– DM, 10 MHz und 8-KByte-Cache-Speicher. **Ihr XT + Comp. schneller als AT!** Hdl.-Anfragen erwünscht! **FSM-Trogisch,** 3051 Suthfeld, Gartenstr. 10, ☎ 0 57 23/8 19 46 Ⓖ

SCHRITTMOTORSTEUERUNG
XYZ-Achsensteuerg. f. Computer mit Parallelport. Mit Netzteil und 3 Schrittmotoren **DM 269**
BOHRPROGR./C64 **DM 98**
Schrittmotor DM 29. Info DM 2. **PME,** Hommerich 20, 5216 Rheidt, ☎ 0 22 08/28 18 Ⓖ

HARDWARE-MESSWERT-ERFASSUNG für fast alle Computer (siehe mc-Quickie). **L. Bockstaller** Ⓖ

NEU IN ANDERNACH: PC-/XT-/AT-Kompatible zum Anfassen und Testen und vieles mehr. Computer-Corner. ☎ 0 26 32/4 31 19 Ⓖ

20-MByte-Festplatte incl. Contr. sowie EPSON-Drucker und IBM-XT-komp. Computer. ☎ 09 21/4 45 20, ab 18 Uhr

Apple II+ Komp., IBM-Geh., Monitor, 2 LW je 640 KByte, Ehring Control, 160 Tr., Tastat., Z80-Karte, CP/M 2,23 Turbo-Pascal, DBase II, MS-Basic, Wordstar, MP, div. Apple- + CP/M-Dienstprogrm., 25 Disc, 160 Tr., kompl. DM 1400.–. ☎ 0 41 54/29 78

KONTAKTE

Suche Verw.-Software für Ärzte zwecks Vertrieb u. Betreuung. Bone Wilhelm, Langörgener Str. 2, 8000 München 90

Modula-II-Club! Gute Clubleistungen. Braun Ludwig, Schwaigerstr. 4, 8425 Neustadt-Geibenst.

Suche Christiani-Lehrgang „Mikroprozessortechnik". Zuschr. unt. Nr. mc 093

PC-5000-Anwender!
Suche Erfahrungsaustausch. G. Hille, ☎ 0 80 24/62 54

HX-20-CLUB und PC-PROFI-GRUPPE c/o Jürgen und Susanne Miersch, Hindenburgd. 58a, 1000 Berlin 45

Microcomputerclub F. PC/AT und CBM. Informat. gegen Freiumschl. PTC, Kantstr. 12, 6050 Offenbach

Freier Mitarbeiter zur Erstellung von Software für Bildverarbeitung und NC-Maschinen gesucht. ☎ 0 95 45/75 23

VERSCHIEDENES

RESTPOSTEN: 500 St. NC-Stab-Akkus (Herst. Saft) 3VR, 1,2RR, 1200 mA/h. **Preis:** DM 5.70/St. 1200 St. Dioden P6KE6.8ARL, **Preis:** DM –.92/St.
☎ 0 60 57/12 66 + 7 65 [G]

mc bis 12/86, DM 70.–.
☎ 0 81 51/1 31 07

TECHNISCHE ÜBERSETZUNG Dipl.-Elektro-Ing. (Englisch) u. techn. Übersetzerin übersetzen günstig Deutsch–Englisch und Englisch–Deutsch.
☎ 0 89/7 25 81 86

DATENTRANSFER zwischen APPLE 35 Tracks und IBM 360 KByte, pro Disc nur 10 DM.
☎ 02 34/28 86 84

Hefte u. kompl. Jahrgänge mc, ELO, Chip, c't. Liste: G. Rubel, Wolf-Eberstein. 10, 7553 Muggensturm

Durchkontaktierte Leiterplatten; Oberfläche Pb/Sn ungeschmolzen, Klein- und Mittelserien, Muster in wenigen Tagen, auch einseitig. **Bestückungen – Layouts.** M. Ernst, Elektron. Systeme, Pf. 8 41, 5800 Hagen 1,
☎ 0 23 31/40 30 07 [G]

Verkaufe 750 Adressen von Computerbesitzern ab Juni 87, Infos: Jürgen Pfeiffer, Langemarckstr. 15, 8390 Passau

Suche zu ID-100 Systems Corp. Unterlagen und EPROMS für die Color-Option. P. Mueri, Zimmermannstr. 25, CH-5200 Brugg

JOYCE → PC, JOYCE → ATARI CP/M → PC, CP/M → ATARI Info: Bernd Drost, Schulstr. 67, 6382 Friedrichsdorf,
☎ 0 61 75/6 04 [G]

8. Nürnberger Flohmarkt für Amateurfunk- und Computerfans, 23. April 1988 v. 8.30–17.00 Uhr im Nürnberger Messezentrum. Private u. profess. Anbieter. Über 5000 qm Ausstellungsfläche. Einheitspreis: DM 20.– p. Tisch. Ausstellerinfo: Hans Kammler, Laurentiusstr. 9, 8500 Nürnberg 60,
☎ 09 11/64 44 34 [G]

Leiterplatten: Sofortdienst für Muster, Layouts, Entwicklungen.
☎ 0 77 02/22 88 u. 18 84 [G]

286-Speed-Karte nur 798.– DM, 10 MHz und 8 KByte Cache-Speicher; Ihr XT+Comp. schneller als AT! Hdl.-Anfragen erwünscht! **FSM-TROGISCH**, Gartenstr. 10, 3051 Suthfeld,
☎ 0 57 23/8 19 46 [G]

∗ BESTÜCKUNG ∗ MONTAGE ∗ PRÜFUNG ∗ Vom Muster bis zur Serienproduktion. Jede Leiterplatte maschinengelötet, sichtkontrolliert und blitzblank ultraschallgereinigt. Fertigung mit oder ohne Material. Ein schnelles Preisangebot erhalten Sie nach Einsendung von Bestükkungsplan und Stückliste oder Bestückungsmuster. gk-electronics ∗ Hofrichterstraße 34, 5000 Köln 80 ∗ ☎ 02 21/63 98 97

Heizungsregler: verbesserte Version der selbstlernenden Hausheizungsregelung aus mc 1/83 bzw. EMUF-Sonderheft, Bausatz komplett mit Gehäuse DM 460.–, Fertiggerät DM 580.–. Ing. Büro Hacker, Am Eichholz 6, 8653 Mainleus,
☎ 0 92 29/10 00 [G]

SCHRITTMOTORSTEUERUNG
XYZ-Achsensteuerg. f. Computer; mit Parallelport. Mit Netzteil und 3 Schrittmotoren **269 DM**
BOHRPROGR./C64 98 DM
Schrittmotor 29 DM. Info 2 DM. **PME,** Hommerich 20, 5216 Rheidt,
☎ 0 22 08/28 18 [G]

Gewerbl. Anzeigen sind mit [G] gekennzeichnet.

STAUBSCHUTZHAUBEN – EIN PROBLEM WENIGER –
z. B. für IBM-PC und Kompatible, zweiteilig (Zentraleinheit mit Monitor und Tastatur).
NUR DM 40.– zzgl. Versandk.
Gesamtpreisl. DM 3.– Briefm.
K & S-VERSAND
K. Schellhammer, Dorfstr. 26, 2061 Kleinwesenberg
☎ **0 45 33/35 66** [G]

Die 2. Auflage des ZX-Hardwarebuches ist da! **Starke Schrittmotore!** Schrittmotor-Steuerung Katalog 6/87 gegen DM 5.– in Briefmarken. Decker & Computer, Pf. 967, 7000 Stgt. 1 [G]

■ DISKETTEN m. Gar.
■ 5¼", 48 tpi, 2 D **DM –.79**
■ 5¼", 96 tpi, 2 F **DM 1.47**
■ 5¼", HD, 1,2–1,6 MB **DM 3.39**
■ 3½", 135 tpi, 2 DD **DM 2.49**
■ 3" Marke Maxell 2er **DM 6.19**
■ Allg. Austro-Agt. Schleißheimer Str. 16, 8057 Eching,
■ ☎ 0 89/3 19 54 56

Werden Sie Ihr eigener Chef
mit einem eigenen, lukrativen Kleinunternehmen. Die besten Branchen, die besten Chancen zeigt neuartige Wirtschaftszeitschrift. X Beispiele mit allen Zahlen, Fakten, Adressen + Tips. Gratisinfo: Die Geschäftsidee, Th.-Heuss-Str. 4/MC801, 5300 Bonn 2









Heft 4 vom	28.3.88
Anzeigenschluß	22.2.88
Heft 5 vom	25.4.88
Anzeigenschluß	21.3.88
Heft 6 vom	30.5.88
Anzeigenschluß	25.4.88

Auftrag für Gelegenheitsanzeige in mc

Anschrift: Franzis-Verlag GmbH, Anzeigenabteilung, Karlstraße 41, 8000 München 2

Name und Adresse ______________________________

______________________________ Unterschrift/Ich zahle sofort nach Rechnungserhalt. ______________ Datum

☐ Preis für private Gelegenheitsanzeige je Druckzeile 4.– DM. Chiffregebühr je Anzeige 8.– DM.

☐ Preis für gewerbliche Gelegenheitsanzeige je Druckzeile 14.82 DM (dürfen nicht unter Chiffre erscheinen). Bitte beachten Sie, daß **Preise in Anzeigen die MwSt.** enthalten müssen.

mc-programmbörse
- ☐ suche Software
- ☐ biete an Software

mc-minimarkt
- ☐ suche Hardware
- ☐ biete an Hardware

- ☐ Tausch
- ☐ Verschiedenes
- ☐ Kontakte
- ☐ Chiffre-Nummer
- ☐ vollständige Adresse
- ☐ Telefon

(Zutreffendes bitte ankreuzen)

Ich wünsche folgenden Text in der nächstmöglichen Ausgabe zu veröffentlichen:

Bitte mit Schreibmaschine oder in Druckschrift in Groß- und Kleinbuchstaben ausfüllen.

Franzis'

Qualifizierte Fachleute finden Sie mit Sicherheit in qualifizierten Fachzeitschriften.

Mit Ihrer Personalanzeige in der mc ...

... erreichen Sie wichtige Beschaffungs-Entscheider aus der Entwicklung und industriellen Anwendung ohne Streuverluste.

... sind Sie nicht nur an einem Wochenende aktuell, sondern so lange das Heft im Umlauf ist.

... sprechen Sie auch Fachkräfte an, die im Augenblick vielleicht noch gar nicht an einen Wechsel denken.

Besonders interessant:

Die Dreierkombination

Funkschau + mc + Elektronik

mit einem Kombinationsrabatt von 35%

Fordern Sie ausführliche Informationen an.
Sprechen Sie mit Heide Mai, Telefon 0 89/51 17-3 41.



Franzis Verlag GmbH
Karlstr. 37–41
8000 München 2
Telex 5 22 301
Telefax 0 89/5117 – 3 79

INSERENTENVERZEICHNIS



Die Mikrocomputer-Zeitschrift

Verlag: Franzis Verlag GmbH, Karlstraße 37–41,
8000 München 2
Postfach 37 01 20, 8000 München 37
Telefon (0 89) 51 17-1
Telex 5 22 301
Telefax (0 89) 51 17-3 79

Alleingesellschafter: G. Franz'sche Buchdruckerei G. Emil Mayer, München. Persönlich haftende Gesellschafter: Peter G. E. Mayer, Verleger (50 %), Michael-A. Mayer, Verleger (50 %); beide wohnhaft in München.

Geschäftsführer:
Peter G. E. Mayer, Michael-Alexander Mayer

Verlagsleitung: Peter Habersetzer, Eugen Wintersberger

Anzeigenleitung: Dietger Kötter

Anzeigenverkaufsleitung: Hans-Joachim Hecht (-3 86)

Disposition: Ingrid Daschner (-2 97)

Anzeigenvertretung Inland:

Norddeutschland: Lita Lange, impulse medien service GmbH, Holtenklinker Str. 9, 2050 Hamburg 80, Tel. 0 40/ 7 24 20 37/38
Hessen: Günter Junne, Victor-Achard-Str. 30, 6380 Bad Homburg v. d. H., Tel. 0 61 72/3 33 94
Bayern: Elfie Rusch, Münchner Verlagsvertretung, Sperberstr. 23, 8000 München 82, Tel. 0 89/4 30 73 32
Nordrhein-Westfalen: Bandelow + Partner GmbH & Co. KG, Herr Werner Bandelow, Dellestr. 36, 4000 Düsseldorf 12, Tel. 02 11/20 14 64
Baden Württemberg: Ulrich G. Felger, Honoldweg 27, 7000 Stuttgart 1, Tel. 07 11/63 27 18
Berlin: Rainer W. Stengel, Bischofsgrüner Weg 91, 1000 Berlin 46, Tel. 0 30/7 74 45 16

Anzeigenvertretung Ausland:

Belgien: ECI/United Media International S.A., Avenue de la folle chanson, 2 bte 7, 1050 Bruxelles, Tel. 02/6 47 31 90, Telex 63 950 eci um
Großbritannien: Martin Geerke, Friary Hall, Flat 3, Firary Road, South Ascot Berks. SL5 9HD, Tel.: 9 90/2 86 49 oder 2 76/6 28 28, Telex 8 58 328
Japan: International Media Rep. Ltd., 2–29, Toranomon 1-chome, Minato-ku, Tokyo 105, Tel. 5 02-06 56, Telex 22 761
USA: DemoNet Inc., 7310 Adams Street, Paramount, CA 90723, Tel. (2 13) 4 08-19 68
Frankreich: Agence Gustav Elm, 41, Avenue Montaigne, 75008 Paris, Tel. 01-47 23 32 67, Telex 2 90 260
Italien: Rancati advertising, Milano San Felice Torre 5, 20090 Segrate, Tel. 02-7 53 14 45, Telex 3 11 250 PP MI I
Schweiz: Exportwerbung AG, Kirchgasse 50, 8024 Zürich, Tel. 01-47 46 90, Telex 8 12 765

Anzeigen nach Preisliste Nr. 8, gültig ab 1. 10. 1987

Vertriebsleitung: Peter Habersetzer

Abonnement: Christa Fischer

Handelsverkauf: Dietlind Stauder

Erscheinungsweise: Die mc erscheint monatlich, jeweils montags am Monatsanfang bzw. am Ende des Vormonats; im 8. Jahrgang

Bezugspreise Inland: Einzelheft 7.– DM, Jahresabonnement 70.– DM; Vierteljahresabonnement 18.50 DM. Studenten, Auszubildende und Rentner erhalten das Jahresabonnement gegen Nachweis günstiger. Der Versand ist im Abonnementspreis eingeschlossen. In den Preisen ist die gesetzliche Mehrwertsteuer in Höhe von 7 % enthalten.

Verlagsvertretungen Ausland:

Belgien: Office International des Périodiques (O.I.P.), Avenue Marnix 30, B-1050 Brüssel – Einzelheft 182.– bfr, Jahresabonnement 1963.– bfr.
Dänemark: Harck + Gjellerups Booksellers Ltd., Fiolstraede 31–33, DK-1171 Kopenhagen K. – Jahresabonnement 316.– dkr.
Frankreich: Librairie Parisienne de la Radio, 43, rue de Dunkerque, F-75010 Paris.
Luxemburg: Messageries Paul Kraus, 5, rue de Hollerich, L-Luxembourg.
Niederlande: De Muiderkring BV, Electronics House, Postbus 313, 1380-AH Weesp – Einzelheft 8.85 hfl, Jahresabonnement 89.50 hfl.
Österreich: Erb-Verlag Ges.m.b.H. & Co. KG, Buch- und Zeitschriften-Vertrieb, Amerlingstr. 1, A-1061 Wien. – Einzelheft 60.– öS, Jahresabonnement 620.– öS.
Schweiz: Verlag Thali AG, CH-6285 Hitzkirch/Luzern – Einzelheft 7.– sfr, Jahresabonnement 67.20 sfr (je nach Kurs).

VORSCHAU



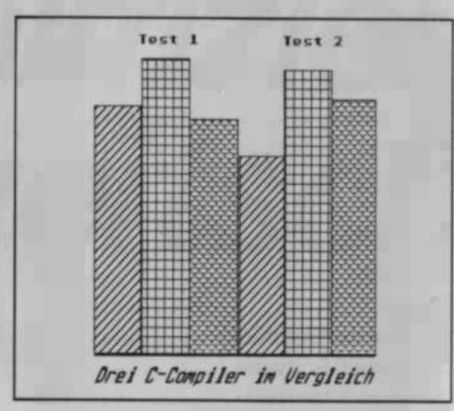
VGA-Karte transparent

Zusammen mit der PS/2-Serie hat IBM eine Grafikkarte mit höherer Auflösung und reicherer Farbauswahl vorgestellt. Einer der größten amerikanischen Hersteller von Grafikkarten für den PC, Video Seven, bietet eine VGA-kompatible Karte an, die kompatibel zu EGA, CGA, MDA und Hercules ist. Wir gehen ausführlich auf die Eigenschaften der Vega VGA genannten Karte ein und zeigen ihre Programmierung.



Karikaturen mit dem Amiga

Wie leistungsfähig das Basic der Amiga-Computer ist, zeigt dieser Beitrag. Aus einem mit der Maus digitalisierten Portrait und einem „Bezugs-Gesicht" errechnet der Amiga eine Karikatur. Nebenbei lernt man eine ganze Menge über Grafik, Screens und ein Basic, das mit seinen FOR- und WHILE-Strukturen schon fast keines mehr ist.



RSA-Algorithmus

So alt wie die Schrift ist der Wunsch, Informationen zu verschlüsseln. Optimale Sicherheit verspricht der RSA-Algorithmus. Bei diesem Verfahren wird mit drei Schlüsseln gearbeitet. Nach einer Erläuterung der mathematischen Grundlagen wird dieser Algorithmus in Pascal programmiert.

C-Compiler

Unter den Hochsprachen für PC nimmt C unbestritten eine Spitzenposition ein. Jede Umsetzung der Standards zeichnet sich durch einige Besonderheiten aus. Wir haben drei C-Compiler für den Atari ST verglichen und Quick-C für MS-DOS-PC getestet. Im Vergleichstest werden viele Hinweise darauf gegeben, wie man Compiler überhaupt vergleichen kann.

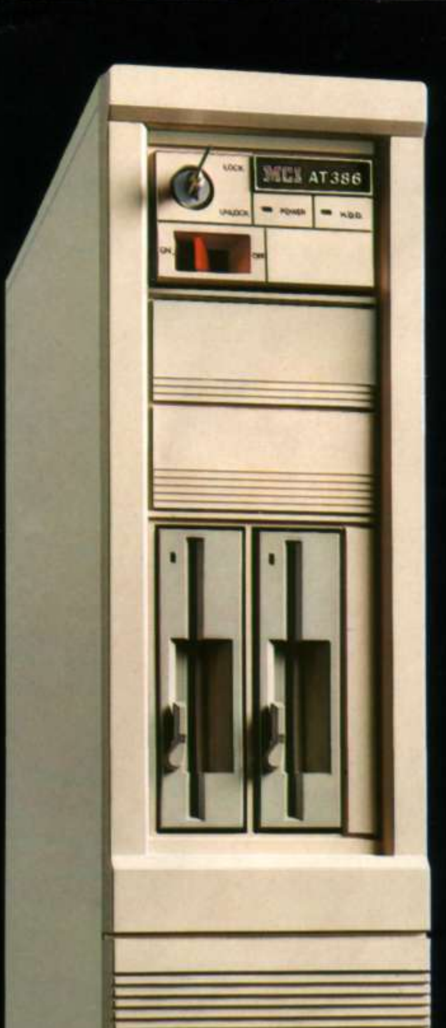


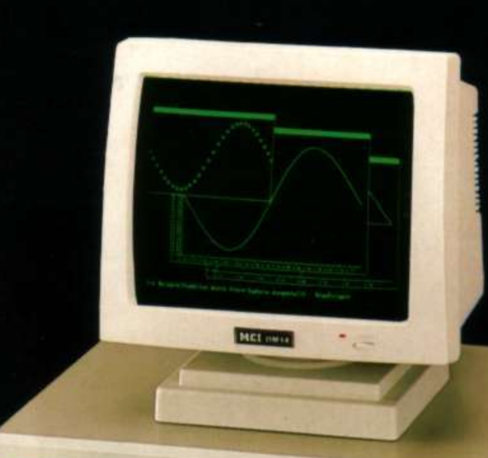
EGA-PC im Test

Der EGA-PC von Schneider und Amstrad weist viele gute Eigenschaften auf. Daß er farblich in die Vollen greifen kann steht in seinem Namensschild. Daß er mit seiner Festplatte und 640 KByte Speicherausbau ein Computer ist, der für einen Einzelarbeitsplatz fast mehr bietet, als sein Besitzer nutzen kann, das wird im Test verraten.

Ausgabe 3/88 erscheint am 22. Februar 1988









Bitte neueste Prospekte anfordern!

